****

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН**

|  |
| --- |
| **Допустить к защите**  Заведующий кафедрой математики и естественнонаучных дисциплин  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г. |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**на тему:**

**«**разработка программы для расчётов и анализа параметров качества изображения, получаемого при съёмке**»**

**Направление подготовки:** 01.03.02. Прикладная математика и информатика

**Профиль:** Программирование, математическое моделирование

**Уровень высшего образования:** бакалавриат

**Форма обучения:** очная

Автор ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Молодов Алексей Алексеевич

подпись (фамилия, имя, отчество)

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н. доцент Раев Олег Николаевич

подпись (учёная степень, учёное звание, фамилия, имя, отчество)

Королёв, 2025

Содержание

[Введение 4](#_Toc201735479)

[1. Анализ математических моделей преобразования оптических изображений светочувствительной матрицей 5](#_Toc201735480)

[1.1. Масштаб изображения 5](#_Toc201735481)

[1.2. Дискретизация изображения по пространственным координатам 7](#_Toc201735482)

[1.3. Квантование сигналов изображения по уровню 8](#_Toc201735483)

[1.4. Выводы 8](#_Toc201735484)

[2. Параметры качества изображения, получаемого при съёмке 10](#_Toc201735485)

[2.1. Пространственная частотная характеристика 10](#_Toc201735486)

[2.2. Резкость изображения 10](#_Toc201735487)

[2.3. Разрешающая способность 10](#_Toc201735488)

[2.4. Искажения, возникающие в результате дискретизации изображения 10](#_Toc201735489)

[2.5. Выводы 10](#_Toc201735490)

[3. Разработка требований к программе 11](#_Toc201735491)

[3.1. Обоснование функций программы 11](#_Toc201735492)

[3.2. Обоснование исходных данных и их диапазонов 13](#_Toc201735493)

[3.3. Обоснование выходных данных 14](#_Toc201735494)

[3.4. Обоснование требований к интерфейсам 14](#_Toc201735495)

[3.5. Выводы 15](#_Toc201735496)

[4. Разработка компьютерной программы 16](#_Toc201735497)

[4.1. Разработка структуры компьютерной программы 16](#_Toc201735498)

[4.2. Разработка интерфейсов программы 17](#_Toc201735499)

[4.3. Подготовка кодов программы 18](#_Toc201735500)

[4.4. Выводы 34](#_Toc201735501)

[5. Разработка программной документации 38](#_Toc201735502)

[5.1. Руководство пользователя 38](#_Toc201735503)

[5.2. Руководство программиста 39](#_Toc201735504)

[5.3. Выводы 41](#_Toc201735505)

[Заключение 42](#_Toc201735506)

[Список источников 43](#_Toc201735507)

[Приложения 44](#_Toc201735508)

# Введение

Цель работы: разработать программу для расчета и анализа параметров пограничной кривой цифрового изображения.

Задачи:

1. Провести анализ математических моделей преобразования оптического изображения, формируемого объективом, и светочувствительной матрицей.
2. Определить параметры качества изображения и обосновать выбор параметра пограничной прямой для оценки и анализа качества изображения.
3. Разработать требования к программе.
4. Реализовать программу с графическим интерфейсом.
5. Разработать программную документацию.

# Анализ математических моделей преобразования оптических изображений светочувствительной матрицей

## Масштаб изображения

Под масштабом изображения в контексте оптической системы понимается отношение размеров изображения объекта на светочувствительном слое к реальным размерам объекта. Это характеристика оптической системы, определяющая, насколько крупным получится изображение на матрице.

При съемке через оптическую систему, такую как объектив, создается изображение объекта. Процесс построения изображения определяется законами геометрической оптики и характеристиками оптической системы, что показано на рисунке 1.

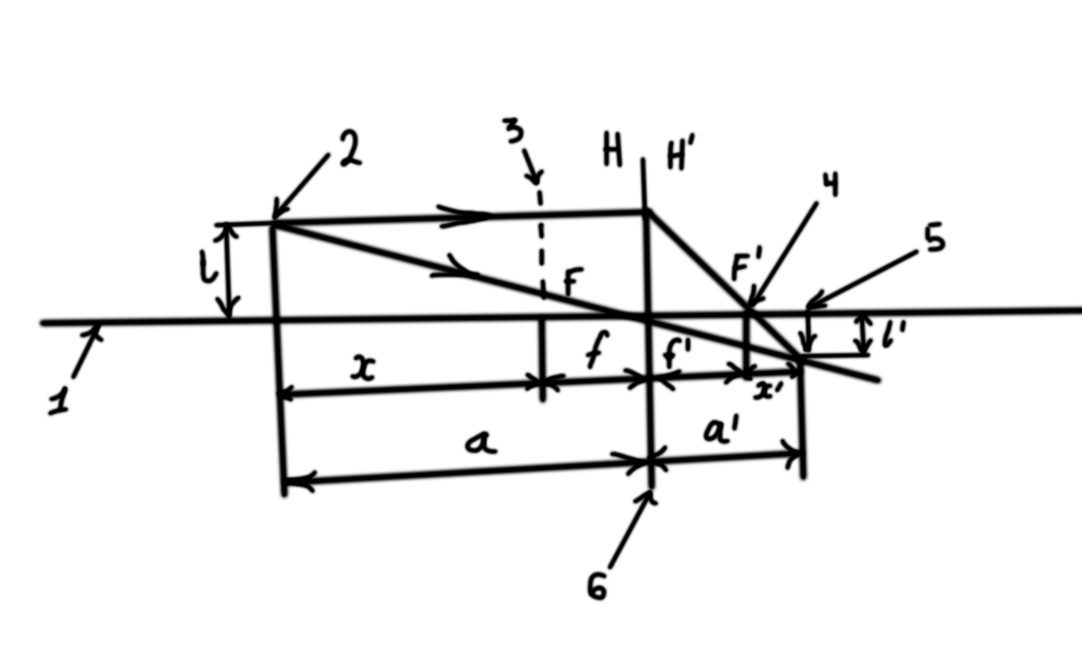


Рисунок 1 – Схема построения изображения объекта

1. Оптическая ось объектива.
2. Объект сьемки.
3. Передний фокус.
4. Задний фокус.
5. Изображение объекта сьемки.
6. Передняя и задняя главные плоскости, совпадающие.

– переднее фокусное расстояние;

– заднее фокусное расстояние;

– размер объекта поперек оптической оси;

– размер изображения.

Если до объектива и после объектива воздух, то *f =* .

Из формулы Ньютона (3) следует, что связь расстояния от объекта до переднего фокуса () и расстояние от заднего фокуса до изображения () выражается через квадрат фокусного расстояния.

Формула Гаусса (4) связывает расстояние от объекта до объектива () с расстоянием от объектива до изображения () через фокусное расстояние объектива (). Если объект находится на разном расстоянии от объектива, изображение будет формироваться на разных расстояниях от объектива. Это приводит к тому, что для объектов, расположенных вне плоскости фокусировки, изображение на светочувствительной матрице будет размытым. Таким образом, резкость изображения зависит от соответствия положения объекта фокусной плоскости.

Формула размера изображения (5) определяет размер изображения () через линейное увеличение, которое пропорционально отношению расстояний ( и .

Светочувствительная матрица представляет собой дискретную двумерную сетку. При фиксированном положении матрицы объекты, находящиеся вне фокуса, проецируются с растянутыми границами. Таким образом, пограничная кривая, представляющая яркостной профиль между участками с разной экспозицией, приобретает определённую ширину. Чем дальше от фокусной плоскости объект – тем шире и менее выделенная граница на изображении.

## Дискретизация изображения по пространственным координатам

Дискретизация изображения – это процесс преобразования непрерывного графического сигнала в дискретную форму.

Изображение, формируемое на светочувствительной матрице, непрерывная по пространственным координатам функция освещённости . При дискретизации она преобразуется в двумерную матрицу значений.

где, – шаг дискретизации;

*n* – номер пикселя по горизонтали;

*m* – номер пикселя по вертикали;

– пространственные координаты.

Согласно теореме Котельникова (7), непрерывный сигнал может быть точно восстановлен из его дискретных значений, если частота дискретизации вдвое больше максимальной частоты сигнала.

где – частота дискретизации;

– максимальная частота сигнала.

Это означает, что шаг дискретизации должен быть мал, чтобы не терялись детали высокой пространственной частоты.

Если теорема Котельникова нарушена, возникают эффекты:

* алиасинг, искажение, при котором высокочастотные детали интерпретируются как низкочастотные;
* муар, периодические искажения, возникающие из-за интерференции между деталями изображения и структурой матрицы.

Эти артефакты связаны с тем, как дискредитируются яркостные переходы. Например, если пограничная кривая узкая, и шаг дискретизации недостаточен – результатом будет “ступенчатый” профиль или артефакты.

## Квантование сигналов изображения по уровню

Квантование по уровню – это процесс преобразования аналоговых значений сигнала в дискретные. Этот процесс необходим для перевода аналоговых дискретных значений освещенности, полученных от фотодиодов, в цифровые значения для компьютерной обработки.

где, *h* – шаг квантования;

*n* – разрядность;

– максимально и минимальное значение освещенности.

Чем выше разрядность, тем точнее представление аналогового сигнала. Квантование приводит к появлению кантовочного шума, заметного на равномерных или плавно меняющихся участках.

## Выводы

Существуют модели преобразования изображения:

геометрическая модель, основанная на законах оптики, определяет масштаб, фокусировку и положение изображения;

частотная модель, основанная на теории обработки сигналов, описывает пространственные частоты и влияние дискретизации.

Обе модели необходимо учитывать при расчётах. Первая – для определения положения и размера объекта на матрице. Вторая – для анализа потерь информации при преобразовании непрерывного изображения в цифровую форму.

Для обеспечения анализа изображения необходимо учитывать искажения, возникающие как на стадии формирования изображения оптической системой, так и на стадии его цифровой регистрации.

# Параметры качества изображения, получаемого при съёмке

## Пространственная частотная характеристика

Пространственная частотная характеристика (ПЧХ) – функция, описывающая, как система передаёт контраст на разных пространственных частотах.

## Резкость изображения

Резкость – определяет коэффициент передачи контраста при малых пространственных частотах.

## Разрешающая способность

Разрешающая способность – это пространственная частота, при которой ПЧХ принимает значение ниже пороговых восприятия человеком.

## Искажения, возникающие в результате дискретизации изображения

## Выводы

# Разработка требований к программе

## Обоснование функций программы

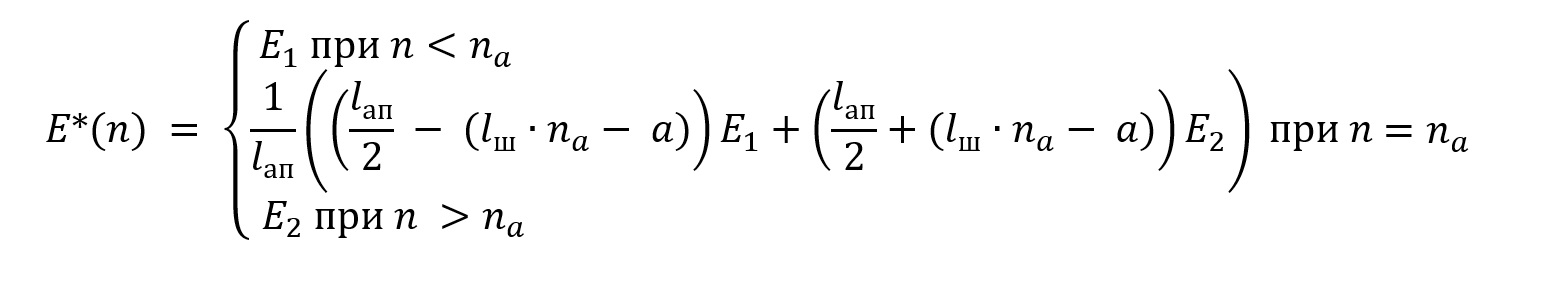
Программа разрабатывается для практических заданий в рамках дисциплины “Обработка изображений” для студентов технических специальностей. Программа должна позволять проводить численные расчёты и визуализировать результаты.

1. Расчет статического края.

Исходные данные: *V* = 0, объектив идеальный, край изображения объекта перпендикулярен оси *x*. Распределение освещенности задано формулой 9.

Исходные данные для расчета: .

Дискретные значения освещенности рассчитываются по формуле 10.



где, – ширина апертуры;

– шаг дискретизации;

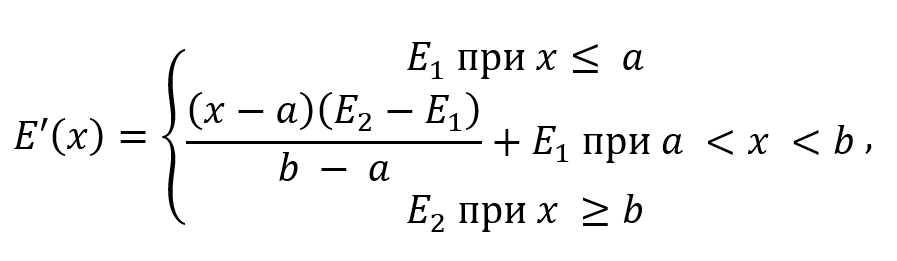
– уровни освещенности;

*a –* положение границы перехода освещенности;

– номер фотодиода, через который будет проходить граница перехода освещенностей.

1. Расчет размытого края.

Исходные данные: *V* = 0; объектив размывает край, ширина размытия краяиз-за аберраций объектива и дифракции света на диафрагмах. Распределение освещенности задано формулой 11.

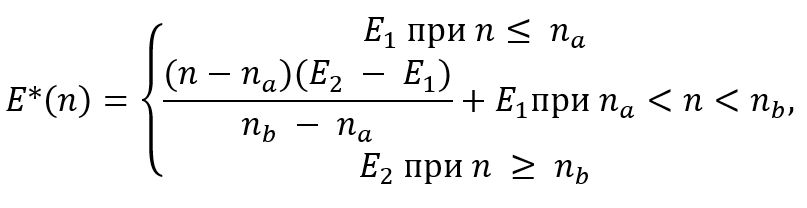


где, *a* – координата на оси *x*, где начинается изменение освещенности;

*b* – координата на оси *x*, где заканчивается изменение освещенности.

Исходные данные для расчета: .

Дискретные значения освещенности рассчитываются по формуле 12.



где, , – номера пикселей, на которые попадают точки перегиба функции .

1. Расчет динамического края.

Исходные данные: движение края происходит вдоль оси *x* со скоростью ; – время экспонирования, обратно пропорционально значению показателя выдержки.

Начальное положение края в момент начала экспонирования *x = a.*

Конечное положение края в момент окончания экспонирования *x = b.* Значение рассчитывается по формуле 13.

Распределение освещенности задано формулой 11.

Дискретные значения освещенности рассчитываются по формуле 12.

1. Построение визуализаций.

Графическое представление дискретных значений освещенности.

Табличное представление дискретных значений освещенности.

1. Интерактивность интерфейса.

Изменение входных параметров.

Изменение способа отображения результатов.

1. Проверка значений допустимым диапазонам.
2. Теоретическая справка.

## Обоснование исходных данных и их диапазонов

Для учебных целей предусматриваются следующие регулируемые параметры представление в таблице 1.

Таблица 1 – Исходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Обозначение | Диапазон | Описание |
| Ширина апертуры |  | 1–10 | Размеры пикселей, мкм |
| Шаг дискретизации |  | |  | | --- | |  |   1–10 | Размеры шага дискретизации, мкм |
| Начальная граница размытия |  |  | Расположение границ в пределах матрицы, мкм |
| Конечная граница размытия |  |  |
| Коэффициент выдержки |  |  | Для расчета *b* |
| Скорость изменения изображения |  |  |
| Освещенность |  | 0 – 100 | Диапазон освещенности, % |
| Размер матрицы |  | 5 – 50 | Для построения матрицы |

## Обоснование выходных данных

Результаты расчётов представляются в нескольких формах, каждая из которых служит своей задаче: анализ дискретных значений и графика распределения освещённости.

1. Матрица освещённости.

Двумерный массив , отображающий распределение освещенности. Используется как для графика, так и для таблицы.

1. Графическое представление.

Отображает структуру перехода освещенности. Цветовая шкала нормирована в процентах.

1. Табличное представление.

Отображает дискретные значения освещенности в каждом пикселе.

## Обоснование требований к интерфейсам

Требования к интерфейсам обосновываются с учётом потребностей пользователей, задач и ожиданий. Программа ориентирована на обучающее применение. Все элементы интерфейса должны учитывать потребности, задачи и ожидания пользователя.

1. Функциональные кнопки, для перехода в соответствующий раздел программы.
2. Использование общей цветовой схемы для интерфейса.
3. Использование черно-белой цветовой схемы для графика.
4. Подписи осей, подпись графика и таблицы.
5. Реакция на некорректные данные, отображение ошибок, пояснения к параметрам.

## Выводы

Программа разрабатывается для численного расчета и визуализации параметра качества изображения – пограничной кривой. Входные параметры и диапазоны подобраны для учебных целей. Выходные параметры включают в себя графические и табличные визуализации, для анализа результатов.

# Разработка компьютерной программы

## Разработка структуры компьютерной программы

Перед разработкой программы были подготовлены функциональные схемы, отражающие архитектуру и детали выполнения заданий, для того чтобы определить последовательность операций.

При запуске программы, открывается главное меню, которое содержит в себе кнопки управления, такие как: “Задание 1”, “Задание 2”, “Задание 3”, “Просмотр теории”, “Просмотр справки”, “Завершение работы”.

При выборе модуля “Просмотр теории”, открывается окно с подробным описанием входных параметров, алгоритмов для решения заданий. Для выхода из модуля предусмотрена кнопка возврата в главное меню.

При выборе модуля “Просмотр справки”, открывается окно с описанием функций кнопок в главном меню. Для выхода из модуля предусмотрена кнопка возврата в главное меню.

При выборе модуля “Завершение работы”, программа завершает работу.

На рисунке 2 представлена обобщенная функциональная схема.

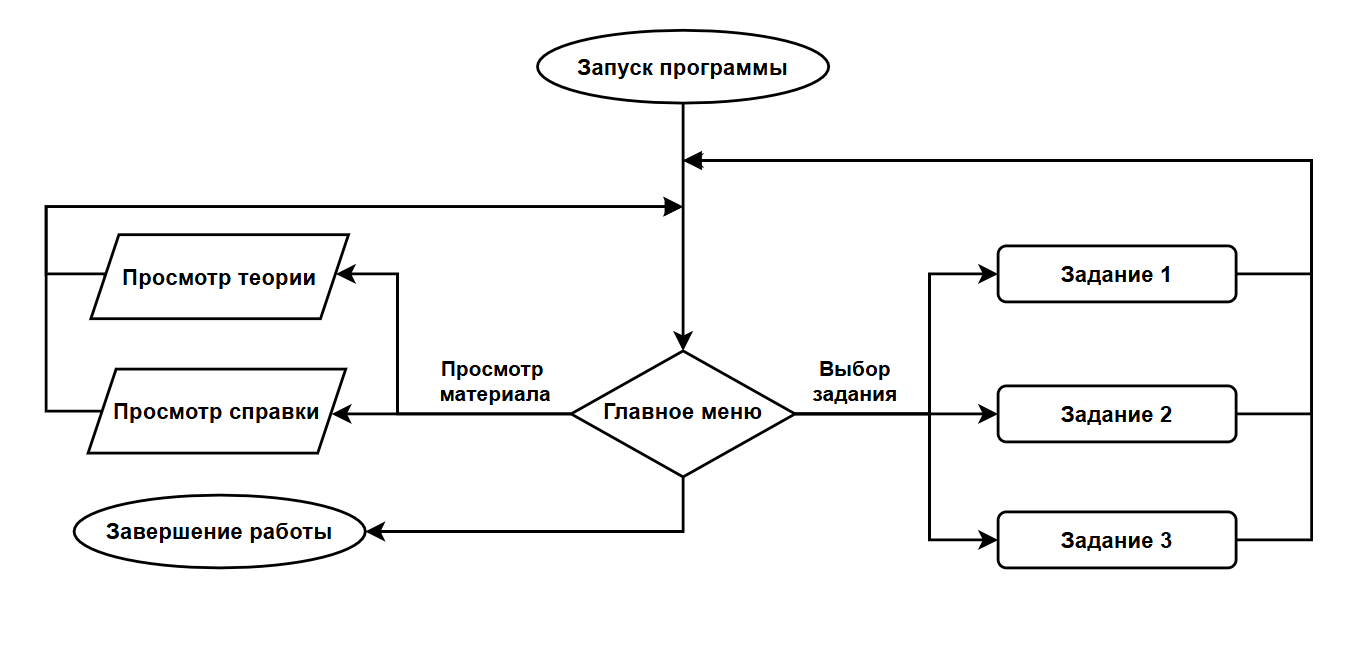


Рисунок 2 – Обобщенная функциональная схема программы

При выборе модулей “Задание 1”, “Задание 2”, “Задание 3”, открывается окно выполнения задания. Для начала выполнения задания пользователь должен ввести параметры, соответствующие допустимым диапазонам, после чего программа проверяет соответствие. Если допущены ошибки, то программа отображает диалоговое окно с указанием параметра, где была допущена ошибка. Иначе программа выполняет расчет, после чего пользователь может выбрать способ отображения результата. Программа предусматривает 2 способа отображения: графический и табличный. Способ отображения можно поменять в зависимости от задачи, которая стоит перед пользователем. После выполнения задания пользователь может заново ввести параметры и продолжить работу или же вернуться в главное меню. Функциональная схема выполнения задания представлена на рисунке 3.

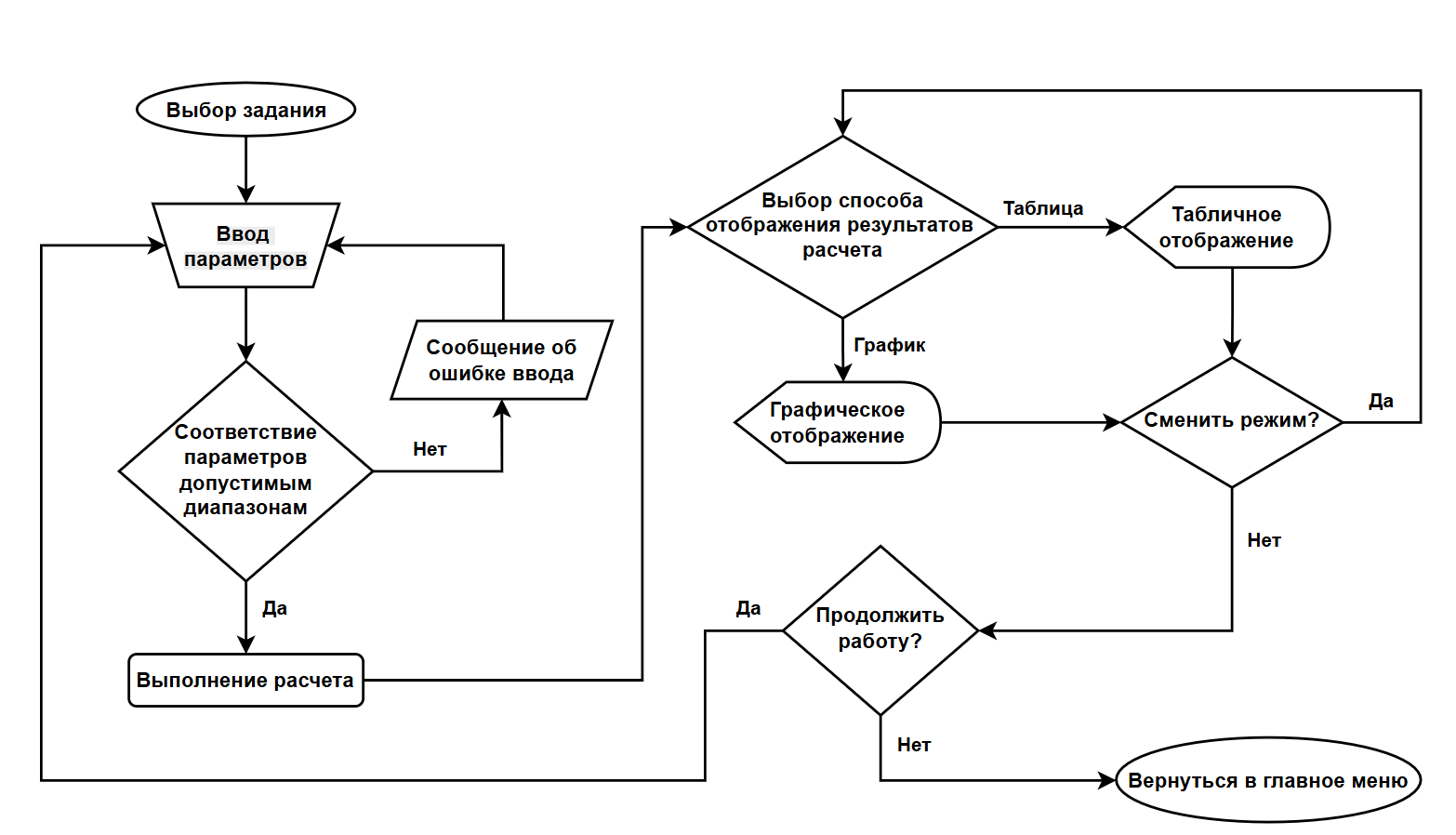


Рисунок 3 – Функциональная схема выполнения задания

Подготовленные функциональные схемы помогают определить логику для дальнейшей реализации программы.

## Разработка интерфейсов программы

Интерфейс разрабатываемой программы должен соответствовать функциональным схемам представленным на рисунках 2 и 3. Быть интуитивно понятен пользователю. Не содержать избыточных элементов. Все элементы должны быть реализованы с единой цветовой схемой.

В программе должен быть единый шрифт, размер шрифта определяется в соответствии с задачей. Например, для модуля “Просмотр теории” шрифт может выделятся для заголовков и подзаголовков.

Для модуля заданий, необходима реакция на ошибки, допускаемые при вводе параметров. Для графического представления необходимо реализовать цветовую шкалу, по которой можно визуально определить значение, отраженное на графике. Для табличного представления необходимо реализовать прокрутку таблицы, если сама таблица больших размеров.

## Подготовка кодов программы

Программа реализована на языке программирования *Python* 3.10.

Используемые библиотеки:

*tkinter* для построения графического пользовательского интерфейса;

*matplotlib* для визуализации результатов;

*numpy* для выполнения численных расчетов;

*ttk* для стилизации интерфейсных элементов.

Программа реализована в виде класса *IlluminationCalculatorApp*, что показано на рисунке 4.

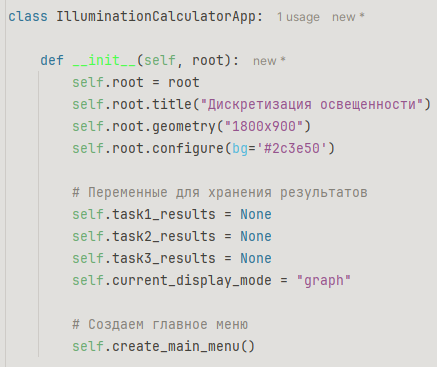


Рисунок 4 – Фрагмент кода программы, класс *IlluminationCalculatorApp*

Внутри конструктора *\_\_init\_\_* выполняются:

установка размера и цвета окна (*root.geometry, root.configure*);

инициализация переменных;

запуск функции создания главного меню *create\_main\_menu*().

Создание главного меню интерфейса происходит в функции *create\_main\_menu*(), это показано на рисунке 5.



Рисунок 5 – Фрагмент кода программы, функция *create\_main\_menu*()

Функция *create\_main\_menu*() содержит кнопки для перехода к:

заданию 1, заданию 2 и заданию 3;

справочному модулю;

теоретическому материалу;

выходу из программы.

Функции *create\_hover\_button*(), *create\_small\_button*() создают кнопки управления, что показано на рисунке 6 и 7.

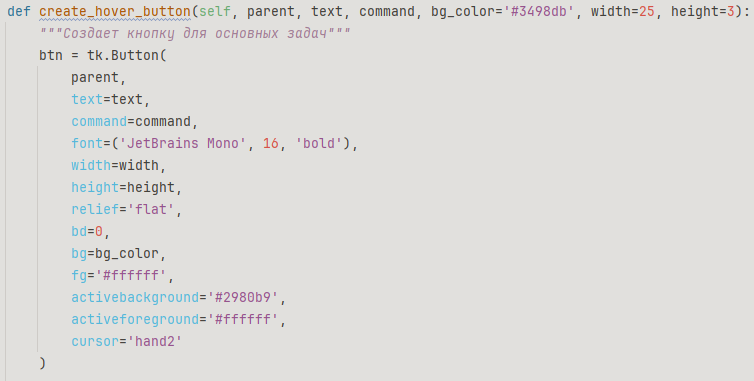


Рисунок 6 – Фрагмент кода программы, функция *create\_hover\_button*()

Функция *create\_hover\_button*(), предназначена для создания кнопки для основных задач. Например, для кнопок главного, кнопки вычислить в окне решения задания.

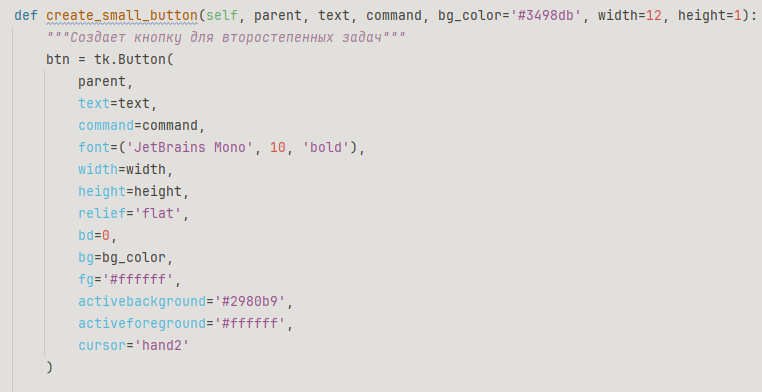


Рисунок 7 – Фрагмент кода программы,

Функция *create\_hover\_button*(), предназначена для создания кнопки для второстепенных задач. Например, для кнопок изменения отображения результатов, кнопки вернуться в окне решения задания.

Функция *create\_help\_button*(), создает кнопку справки, что показано на рисунке 8.



Рисунок 8 – Фрагмент кода программы, функция *create\_help\_button*()

Функции *show\_task1\_page*(), *show\_task2\_page*(), *show\_task3\_page*() загружают интерфейс для выполнения соответствующего задания, что показано на рисунке 9.



Рисунок 9 – Фрагмент кода программы, функции *show\_task\_page*()

Все функции *show\_task\_page*() вызывают обобщенную функцию *create\_task\_page*(*task\_num*), передавая номер задачи.

Функция *create\_task\_page*(*task\_num*) создает структуру окна задачи, что показано на рисунке 10.



Рисунок 10 – Фрагмент кода программы, функция *create\_task\_page*(*task\_num*)

Структура функции *create\_task\_page*(*task\_num*):

левая панель содержит параметры и справку (create\_left\_panel);

правая панель отображает результаты (create\_right\_panel);

верхняя панель содержит кнопку возврата в главное меню.

Функция *create\_left\_panel*() формирует левую панель, для работы с параметрами, которая занимает экрана, что показано на рисунке 11.



Рисунок 11 – Фрагмент кода программы, функция *create\_left\_panel*()

Для каждого задания создаются словари хранения полей ввода, задаются стандартные значения, например, для “*l\_ap*” = “5.0”. Так же в левой панели ниже представлен блок со справкой который выводит обозначения и диапазоны для каждого параметра задачи.

Функция *create\_right\_panel*(), формирует правую панель, для работы с результатами, которая занимает экрана, что показано на рисунке 12.



Рисунок 12 – Фрагмент кода программы, функция *create\_right\_panel*()

Функция отображает кнопки переключения отображения результатов: “График” и “Таблица”, а также область *results\_frame*, где будут визуализированы графическое и табличное представление дискретных значений освещенности.

Обобщающая функция *calculate\_task*(*task\_num*), которая вызывает расчет конкретной задачи, что показано на рисунке 13.



Рисунок 13 – Фрагмент кода программы, функция *calculate\_task*(*task\_num*)

Функция вызывает задания:

calculate\_task1() – статический край;

calculate\_task2() – размытый край;

calculate\_task3() – динамический край.

Так же в этой функции предусмотрено в случае ошибки ввода параметров выводить окно с сообщением об ошибке.

Функции *calculate\_task1*(), *calculate\_task2*(), *calculate\_task3*(), получают параметры и производят расчет для заданий, что показано на рисунках 14–16.

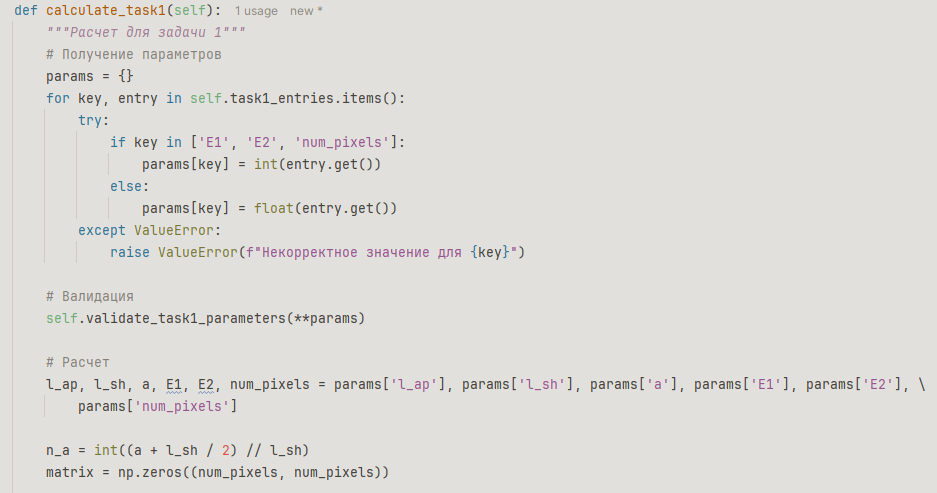


Рисунок 14 – Фрагмент кода программы, функция *calculate\_task1*()

Функция *calculate\_task1*() рассчитывает дискретные значения для каждого пикселя в соответствии с формулой 10.

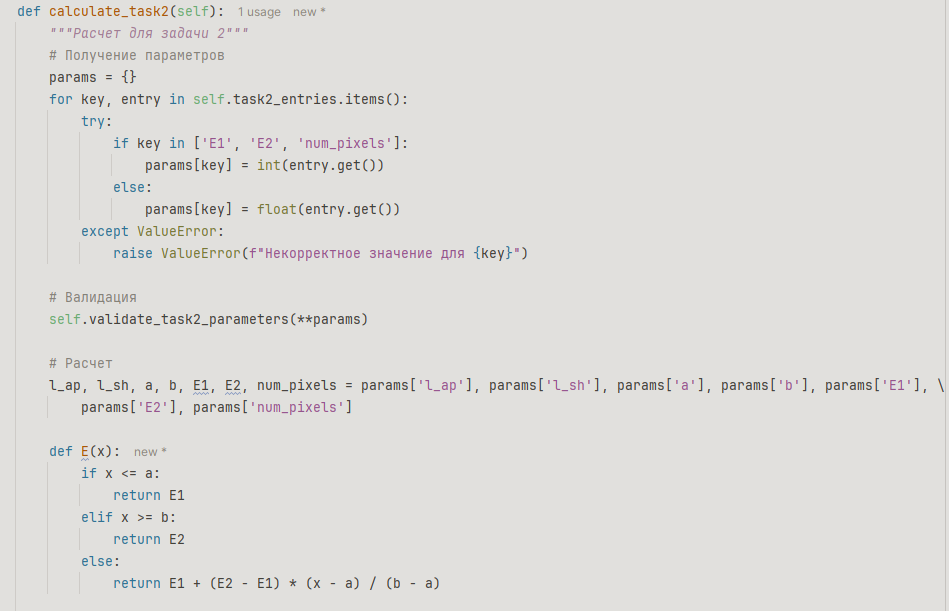


Рисунок 15 – Фрагмент кода программы, функция *calculate\_task2*()

Функция *calculate\_task2*() рассчитывает дискретные значения для каждого пикселя в соответствии с формулой 12.

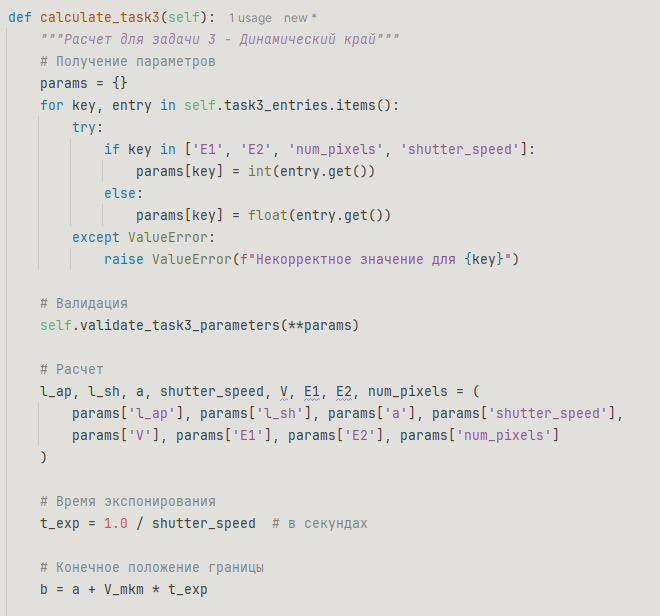


Рисунок 16 – Фрагмент кода программы, функция *calculate\_task3*()

Функция *calculate\_task3*() рассчитывает значение , после чего рассчитывает конечное положение границы *b* по формуле 13. После этого рассчитывает дискретные значения для каждого пикселя в соответствии с формулой 12.

Функция *validate\_task\_parameters*() проверяет допустимые диапазоны параметров, в случае недопустимого значения выдает ошибку по определенному параметру, что показано на рисунке 17.

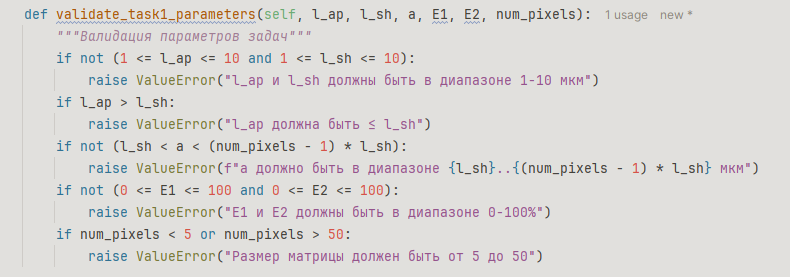


Рисунок 17 – Фрагмент кода программы, функция *validate\_task\_parameters*()

Проверяются диапазоны значений, логические условия, например:

*l\_ap ≤ l\_sh*, *a < b*, *E1 ≤ 100*.

Функция *switch\_display\_mode*() определяет режим отображения результатов, что показано на рисунке 18.

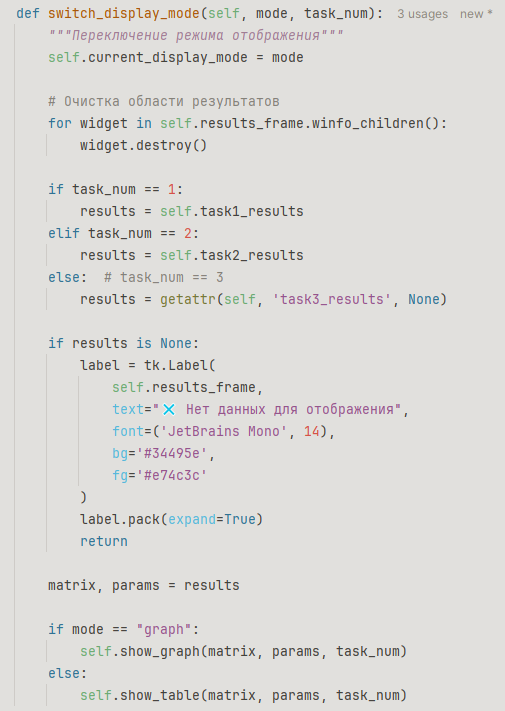


Рисунок 18 – Фрагмент кода программы, функция *switch\_display\_mode*()

Перед отрисовкой очищает предыдущие данные и в зависимости от выбранного режима строит график или таблицу.

Функция *show\_graph*() строит график дискретных значений освещенности, что показано на рисунке 19.



Рисунок 19 – Фрагмент кода программы, функция *show\_graph*()

Также функция *show\_graph*() дополнительно строит шкалу освещенности, для того чтобы пользователь наглядно смог увидеть дискретное значение освещенности.

Функция *show\_table*() формирует таблицу дискретных значений освещенности, что показано на рисунке 20.



Рисунок 20 – Фрагмент кода программы, функция *show\_table*()

Для больших таблиц, например 50х50, предусмотрена прокрутка. Также значения освещенности отображаются с округлением до 1 знака после запятой.

Функция *show\_theory\_page*() формирует окно теоретических сведений, что показано на рисунке 21.



Рисунок 21 – Фрагмент кода программы, функция *show\_theory\_page*()

Вспоминающая функция *get\_theory\_sections*() заполняет окно с теоретической справкой в структурированном виде, что показано на рисунке 22.

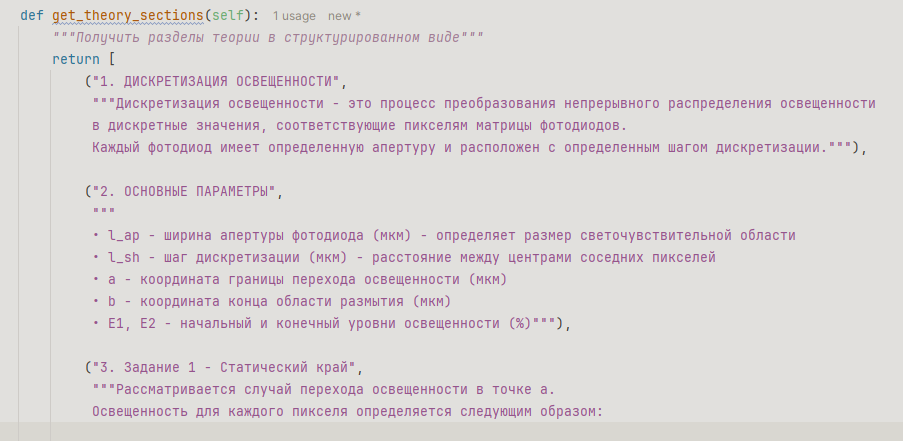


Рисунок 22 – Фрагмент кода программы, функция *get\_theory\_sections*()

Функция *show\_help*() отображает справочное окно с кратким описанием работы программы, что показано на рисунке 23.

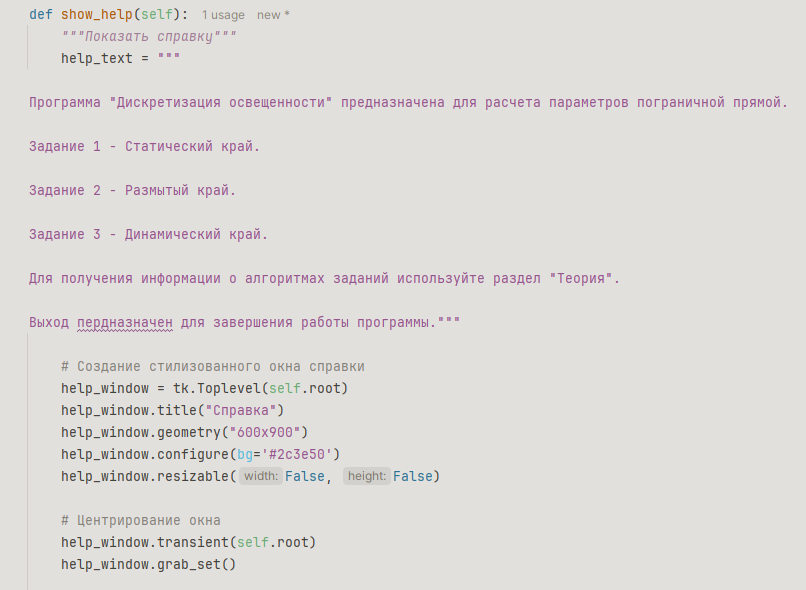


Рисунок 23 – Фрагмент кода программы, функция *show\_help*()

Функция *main*() точка входа в программу, что показано на рисунке 24.

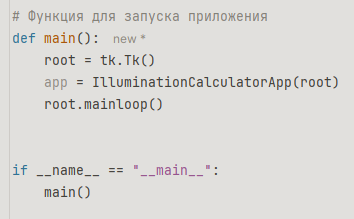


Рисунок 24 – Фрагмент кода программы, функция *main*()

Создаёт объект *tk.Tk*() и запускает главный цикл интерфейса.

## Выводы

Реализована программа для расчета параметров пограничной кривой на языке программирования Python с графическим интерфейсом.

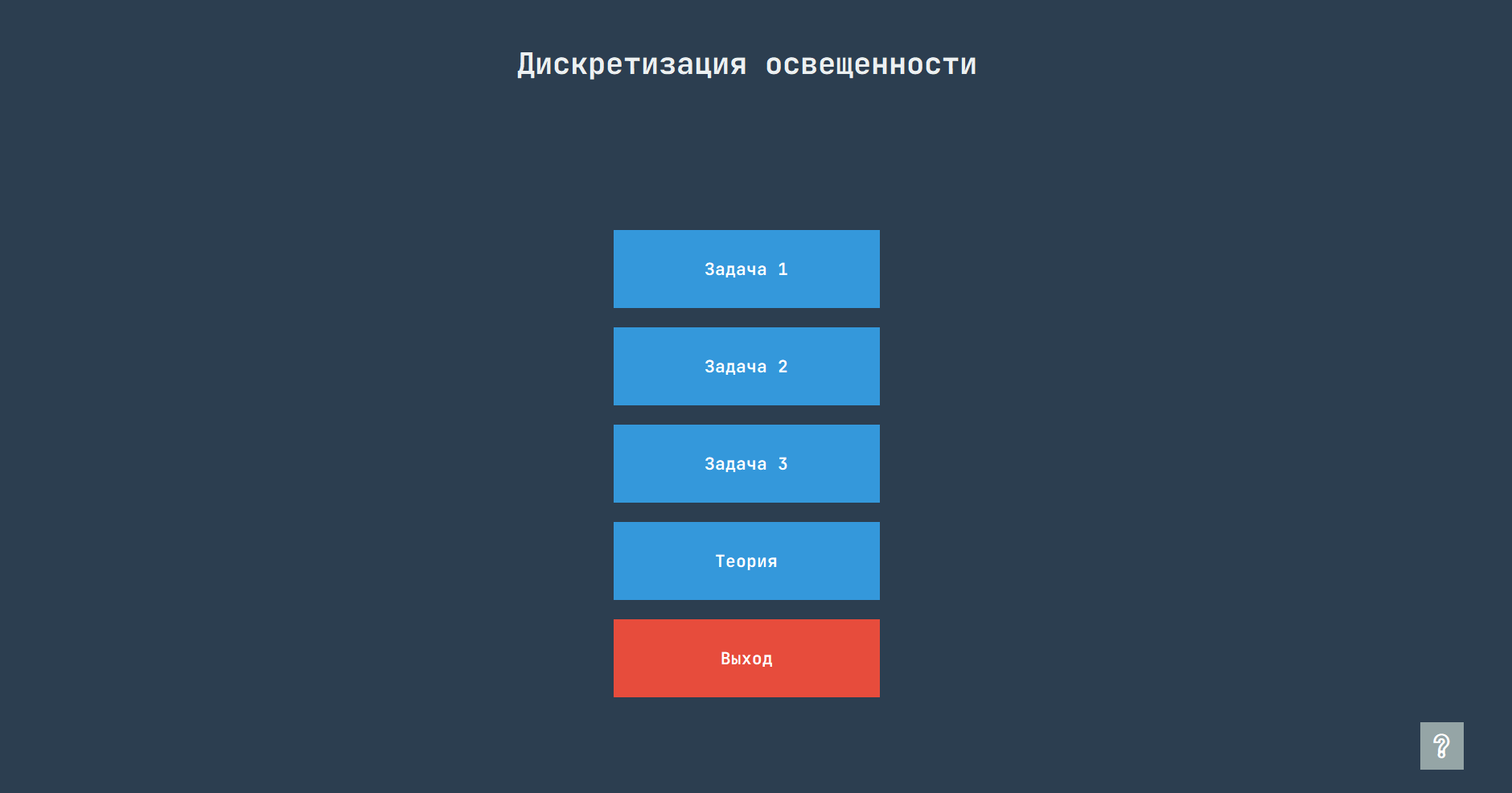


Рисунок 25 – Интерфейс главного меню

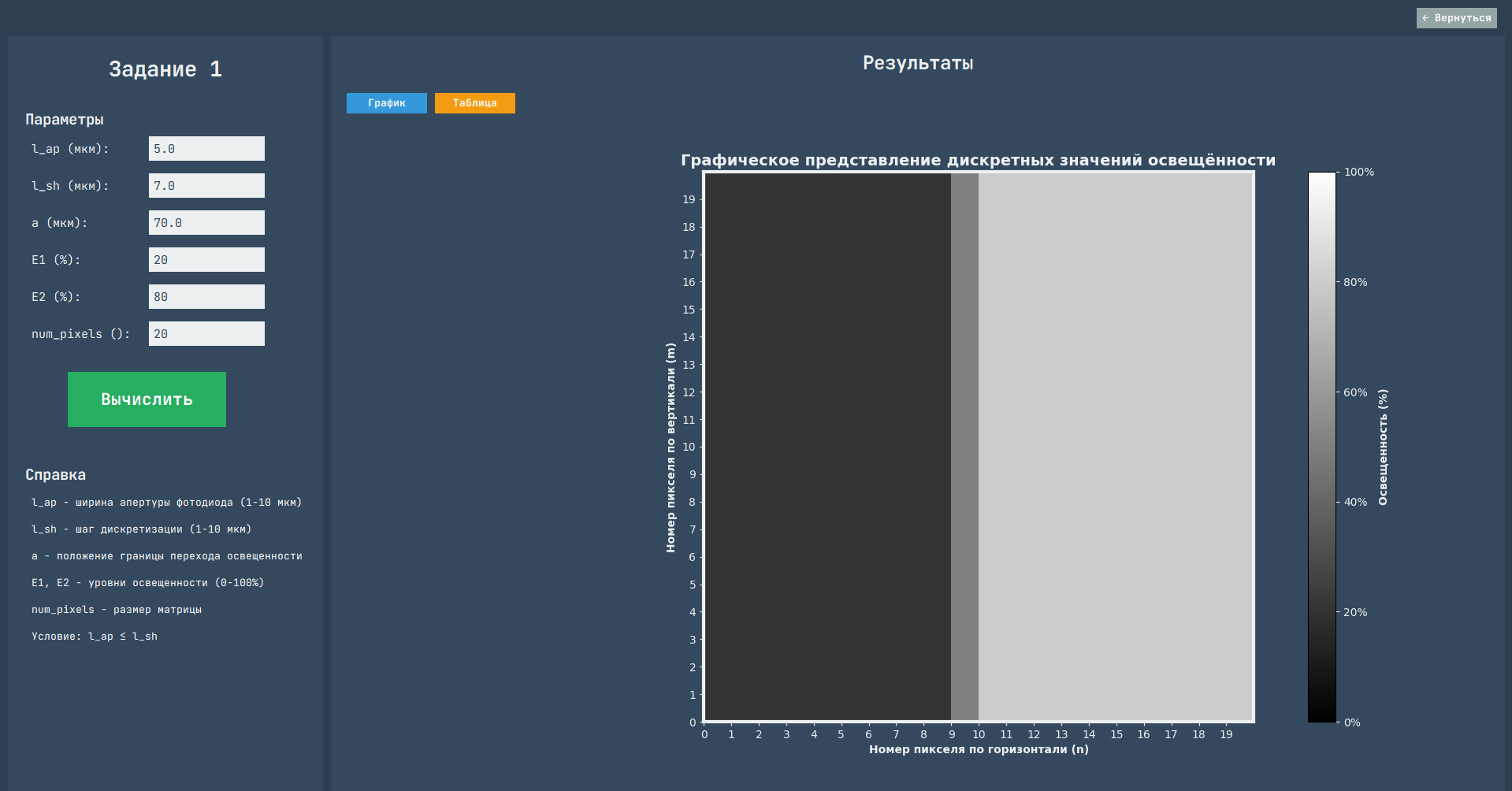


Рисунок 26 – Интерфейс задания 1

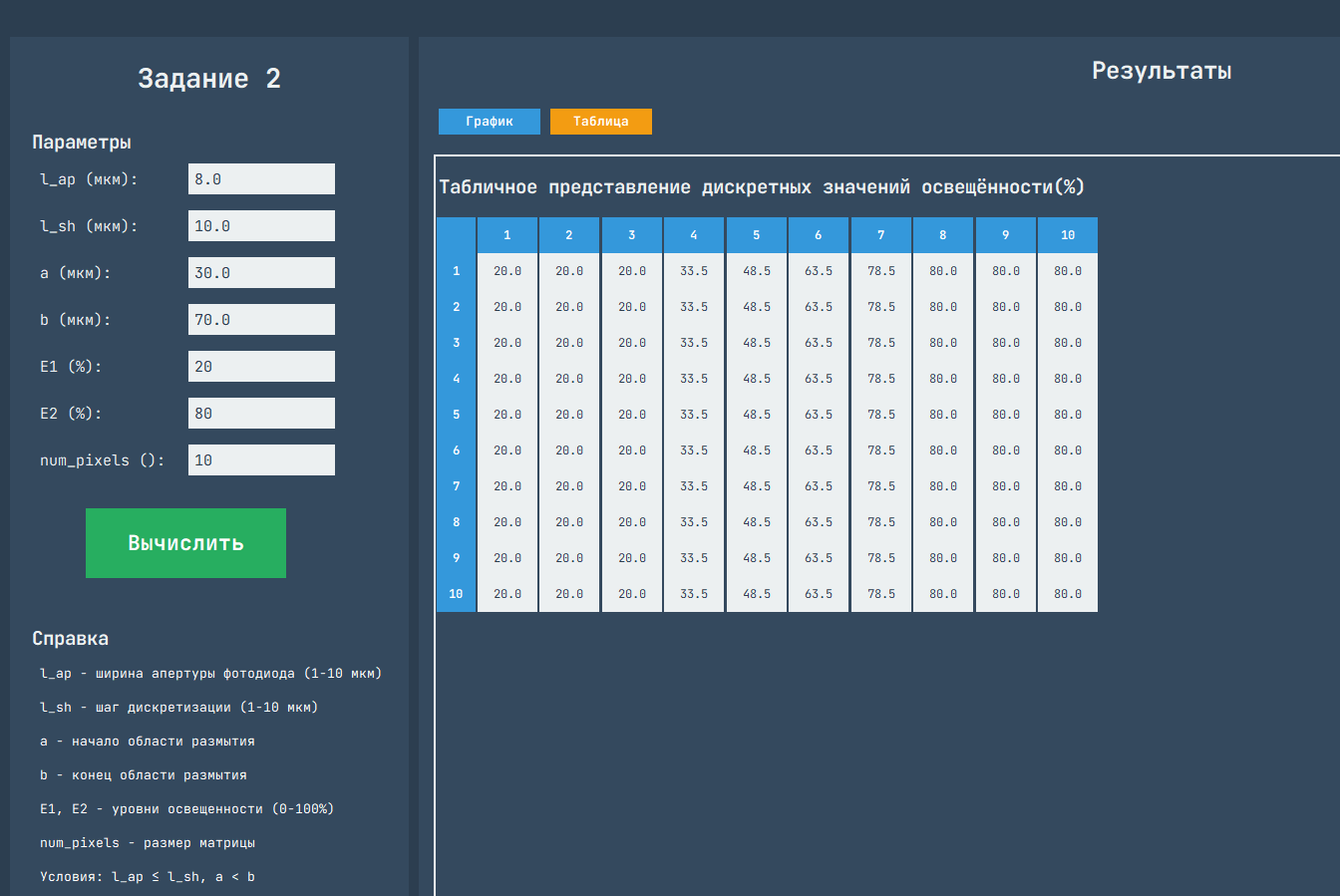


Рисунок 27 – Интерфейс задания 2

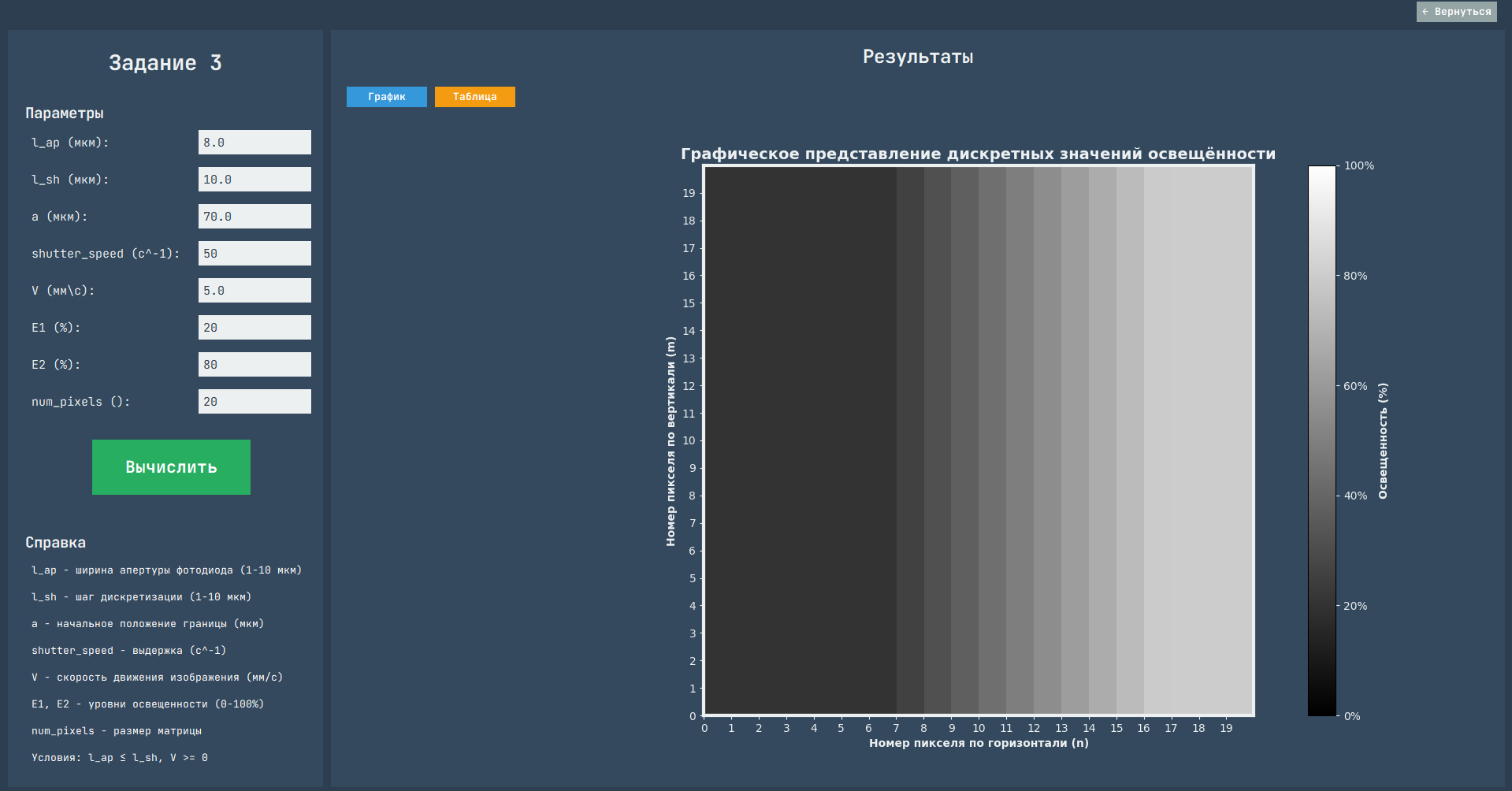


Рисунок 28 – Интерфейс задания 3

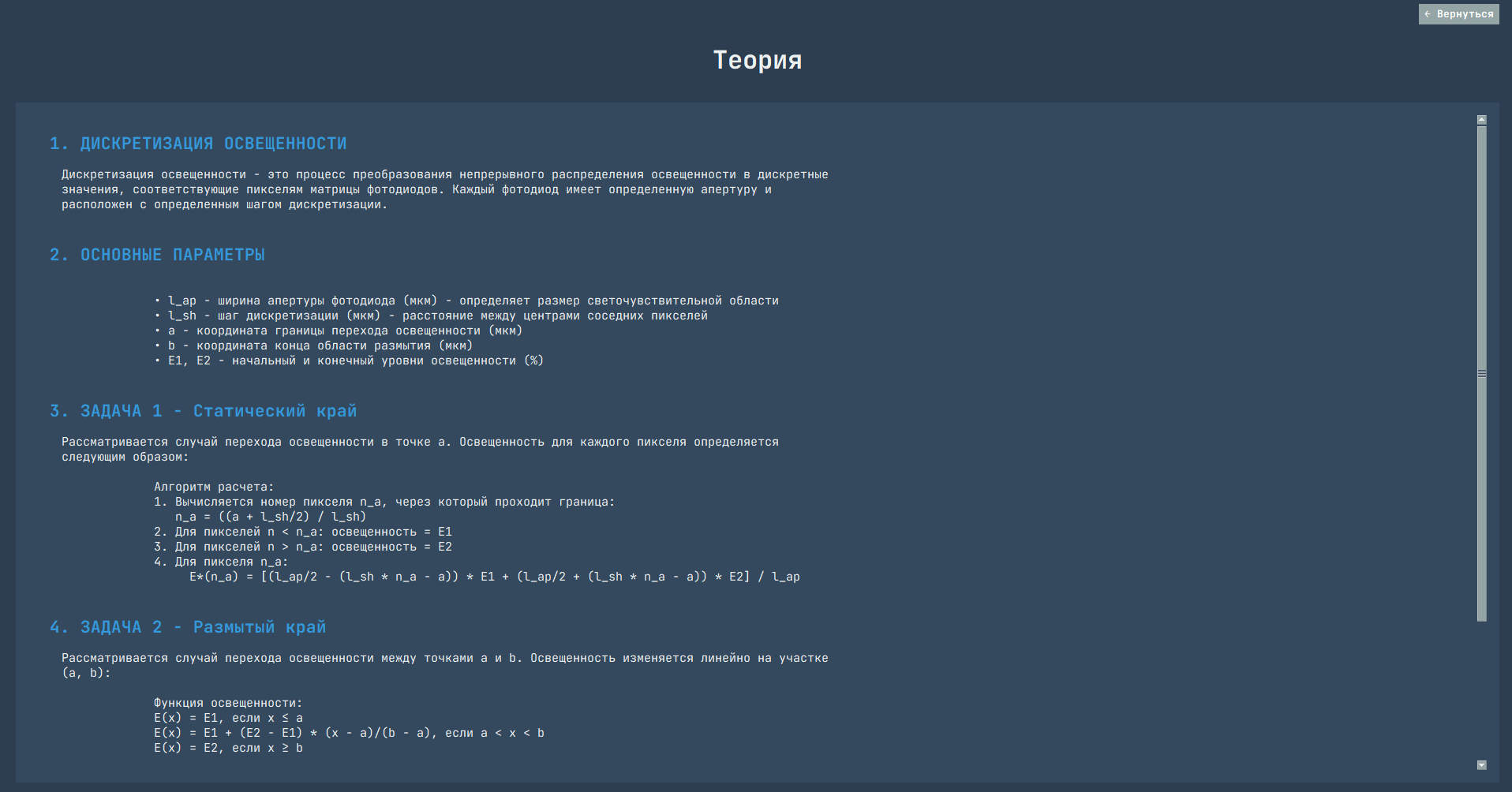


Рисунок 28 – Интерфейс теории

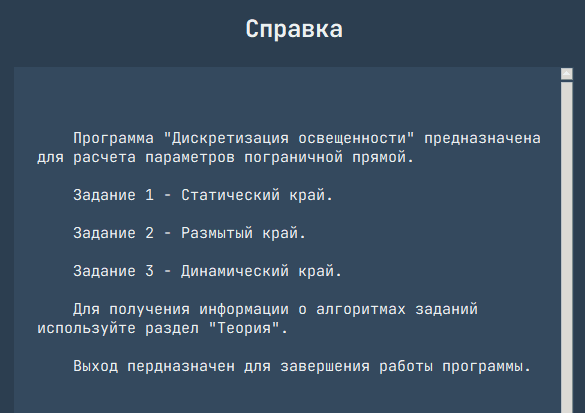


Рисунок 29 – Интерфейс справки

# Разработка программной документации

Программная документация необходима для эксплуатации и сопровождения программы. Она включает инструкции для пользователей (студентов, преподавателей), и для разработчиков, обеспечивая понимание логики работы и структуры кода.

## Руководство пользователя

Руководство пользователя предназначено для лиц, эксплуатирующих программу без необходимости вносить изменения в её код. Программа ориентирована на применение в учебном процессе – для практикумов в рамках дисциплины “Обработка изображений”.

1. Запуск программы.

Выполняется через файл *illumination.exe*.

1. Главное меню.

Кнопка задание 1 – перемещает на страницу задания 1 “статический край”.

Кнопка задание 2 – перемещает на страницу задания 2 “размытый край”.

Кнопка задание 3 - перемещает на страницу задания 1 “динамический край”.

Кнопка теория – перемещает на страницу с теорией для программы, где представлены: алгоритмы, входные и выходные параметры для заданий.

Кнопка справка – отображение назначений кнопок.

Кнопка выход – завершение работы программы.

1. Решение заданий.

Ввод параметров в соответствующие диалоговые окна, диапазоны и описание представлены в таблице 1.

Кнопка вычислить – проверяет диапазоны в соответствии

с заданными, решает в соответствии с алгоритмом задания, возвращает матрицу освещенности.

Кнопка график – отображает графическое представление дискретных значений освещенности.

Кнопка таблица – отображает табличное представление дискретных значений освещенности.

Кнопка вернуться – возвращает в главное меню.

## Руководство программиста

Руководство предназначено для лиц, желающих модифицировать, расширять или сопровождать код программы. В частности, полезно студентам, использующим программу в исследовательских целях. Методы и их описание представлены в таблице 2.

* Язык реализации Python 3.10.
* Библиотеки:

*tkinter* для графического пользовательского интерфейса;

*matplotlib* для визуализации;

*numpy* для вычислений;

*ttk* для стилизации.

* Основной класс: *IlluminationCalculatorApp*.

Таблица 2 – Методы программы

|  |  |
| --- | --- |
| Метод | Описание |
| *create\_main\_menu* | Создает главное меню. |
| *create\_hover\_button* | Создает кнопку для основных задач. |
| *create\_help\_button* | Создает кнопку справки. |
| *create\_small\_button* | Создает кнопку для второстепенных задач. |
| *show\_task\*\_page* | Показывает страницу задачи (\* - номер задачи). |
| *create\_task\_page* | Создает страницу задачи. |
| *create\_left\_panel* | Создаёт левую панель с параметрами и справкой. |
| *create\_right\_panel* | Создает правую панель для отображения результатов. |
| *get\_help\*\_text* | Возвращает текст справки для задачи (\* - номер задачи). |
| *calculate\_task\** | Выполняет расчет для задачи (\* - номер задачи). |
| *validate\_task\*\_parameters* | Проверяет параметры допустимым диапазонам задачи (\* - номер задачи). |
| *show\_graph* | Отображает график. |
| *show\_table* | Отображает таблицу. |
| *switch\_display\_mode* | Переключает режим отображения. |
| *show\_theory\_page* | Отображает страницу теории. |
| *get\_theory\_sections* | Возвращает текст теории. |
| *show\_help* | Отображает справку. |
| *clear\_window* | Отчищает окно. |
| *main* | Основная функция для запуска приложения. |

Для добавления нового задания необходимо расширить метод *create\_task\_page*() и добавление новой *calculate\_task\**() функции.

Все интерфейсные элементы реализованы с единой цветовой схемой и шрифтами. Шрифт “*JetBrains Mono*” используется во всей программе.

Цветовая схема:

* фон: #2c3e50 (темно-синий);
* панели: #34495e (серо-синий);
* кнопки главного меню: #3498db (синий);
* текст: #ecf0f1 (светло-серый);
* кнопка вычислить: #27ae60 (зеленый);
* кнопка выхода: #e74c3c (красный).

## Выводы

Документация разработана для пользователя и программиста.

Руководство пользователя описывает работу программы для решения заданий.

Руководство программиста описывает внутреннюю архитектуру, для работы с программой.

# Заключение

# Список источников

# Приложения