

СОЗДАНИЕ ПРИЛОЖЕНИЯ «СИМУЛЯЦИЯ ОТРАЖЕНИЯ И ПРЕЛОМЛЕНИЯ СВЕТА» НА C# WINFORMS

Е.П. Теслева², к.ф.-м.н, доц., Е.А. Ткачев¹, ученик 9 класса
¹МБОУ «Лицей города Юрги»

652055, Кемеровская обл., Юрга, ул. Кирова, 7

²Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

Аннотация: Статья описывает процесс создания приложения для симуляции отражения и преломления света с использованием языка программирования C# и платформы Windows Forms (WinForms). Приложение предназначено для визуализации взаимодействия света на границе двух сред, имеющих разные коэффициенты преломления. В статье рассмотрены основные шаги разработки.

Ключевые слова: Windows Forms Application, C#, разработка приложения, физика, отражение света, преломление света, VisualStudio.

Abstract: The article describes the process of creating an application for simulating the reflection and refraction of light using the C# programming language and the Windows Forms platform (WinForms). The application is designed to visualize the interaction of light at the boundary of two media having different refractive coefficients. The article discusses the main development steps.

Keywords: Windows Forms Application, C#, application development, physics, light reflection, light refraction, Visual Studio.

Понимание поведения света при взаимодействии с различными средами является важной частью физики. Симуляция отражения и преломления света позволяет более наглядно иллюстрировать эти принципы. В данной статье представлен пошаговый процесс создания приложения для симуляции отражения и преломления света с использованием языка программирования C# и платформы Windows Forms (WinForms). Приложение предназначено для визуализации поведения света на границе между двумя средами с разными коэффициентами преломления.

Цель данной работы заключается в разработке и реализации приложения для симуляции отражения и преломления света с использованием языка программирования C# и платформы Windows Forms (WinForms).

В программе использовались следующие методы:

Метод обработчика события нажатия на кнопку `btnSimulate``:

1. Сначала проверяется, успешно ли удалось преобразовать введенные пользователем значения для угла падения (`txtAngleOfIncidence``), коэффициента преломления первой среды (`txtRefractiveIndex1``) и коэффициента преломления второй среды (`txtRefractiveIndex2``) в тип данных `float``.

2. Если все три значения успешно преобразованы и введенный угол падения не превышает 90 градусов, то вызывается метод `SimulateLightBehavior``, передавая ему в качестве параметров угол падения, коэффициент преломления первой и второй сред.

3. Если введенный угол падения больше 90 градусов, то выводится сообщение об ошибке с просьбой ввести угол падения меньше 90 градусов.

4. Если введенные значения не удалось успешно преобразовать в тип `float``, то также выводится сообщение об ошибке с просьбой ввести корректные значения.

Этот метод предназначен для обработки ввода и запуска симуляции поведения света на границе двух сред.

Метод `SimulateLightBehavior``, который используется для расчета угла отражения и угла преломления света на границе двух сред:

1. Угол падения, коэффициент преломления первой среды и коэффициент преломления второй среды передаются в качестве параметров функции.

2. Угол падения переводится из градусов в радианы, затем рассчитывается угол отражения (равный углу падения, но с противоположным знаком).

3. С помощью закона Снеллиуса вычисляется синус угла преломления.

4. Если $\sin$ угла преломления превышает 1 (что теоретически невозможно), или меньше – 1 (что означает полное отражение), то угол преломления устанавливается в NaN.

5. В противном случае для угла преломления рассчитывается обратное синусное значение.

6. Углы отражения и преломления переводятся обратно из радиан в градусы.

7. Значения угла отражения и угла преломления (если они были рассчитаны) устанавливаются в соответствующие текстовые поля `txtAngleOfReflectionResult``.

Этот метод является частью функционала симуляции поведения света на границе двух сред и позволяет рассчитать углы отражения и преломления в соответствии с законами оптики.

Метод 'DrawRays', который используется для визуализации лучей света и их отражения/преломления на границе двух сред:

1. Создается объект 'Bitmap' для отображения графики на контроле 'pictureBox' заданного размера.
2. Создается объект 'Graphics' для рисования на ранее созданном 'Bitmap'.
3. Фон изображения очищается в белый цвет.
4. Вычисляется центр изображения 'pictureBox'.
5. Вызывается функция 'DrawBoundary', которая рисует границу между двумя средами с заданными коэффициентами преломления.
6. Вызывается функция 'DrawPerpendicular', которая рисует перпендикуляр к границе сред из центра изображения.
7. Вызывается функция 'DrawArrow' для рисования падающего луча (красный цвет), отраженного луча (синий цвет) и, если угол преломления рассчитан, преломленного луча (зеленый цвет).
8. Результат рисования сохраняется в 'pictureBox'.

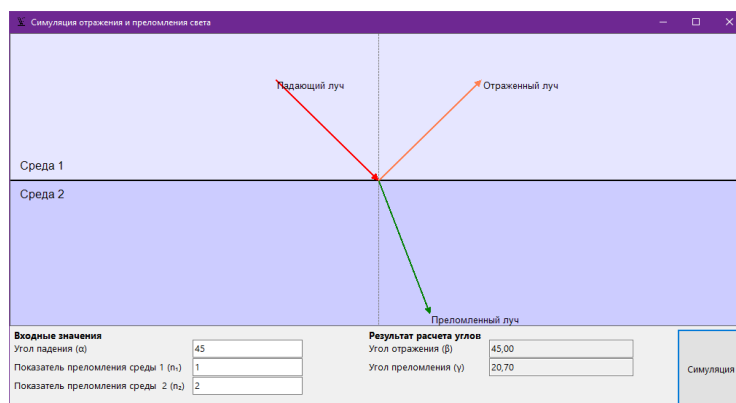


Рис. 1. Основной интерфейс программы

Этот метод предназначен для визуализации процесса отражения и преломления света на границе двух сред и позволяет наглядно отобразить перемещение лучей света при изменении угла падения и коэффициентов преломления.

Метод 'DrawBoundary', который рисует границу между двумя средами с заданными коэффициентами преломления на графике:

1. Создается объект 'Pen' с черным цветом и толщиной линии 2.
2. Вычисляются цвета 'color1' и 'color2' для двух разных сред на основе их коэффициентов преломления. Цвета делаются светлее в зависимости от значений коэффициентов преломления.
3. Создается 'SolidBrush' с цветом 'color1', и используя его, закрашивается верхняя половина графика для первой среды.
4. Создается 'SolidBrush' с цветом 'color2' и закрашивается нижняя половина графика для второй среды.
5. Рисуются линия границы между двумя средами посередине изображения.

Этот метод визуализирует границу между двумя средами на графике, используя цвета, зависящие от коэффициентов преломления каждой среды, и разделяя график на две части для наглядного отображения перехода света между средами.

Метод 'DrawArrow', который рисует стрелку на графике с заданными параметрами:

1. Параметры метода включают графический объект 'g', начальную точку 'startPoint', угол поворота 'angle', цвет стрелки 'color', подпись 'label', флаги 'isRefracted', 'isIncident' и 'isReflected'.
2. Вычисляется длина стрелки как 200 единиц.
3. В зависимости от флагов 'isRefracted', 'isIncident' и 'isReflected', вычисляются конечные координаты стрелки. Если это преломленный луч, координаты рассчитываются в соответствии с углом. Если это падающий луч, направление меняется.
4. После вычисления координат, рисуется стрелка от начальной точки до конечной точки с использованием заданного цвета 'color' и подписи 'label'.
5. Функция рисует стрелки для различных типов лучей (падающий, отраженный, преломленный) в зависимости от переданных параметров и угла поворота.

Список использованных источников:

1. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие / С.Р. Гуриков. – Москва: ФОРУМ: ИНФРА-М, 2020. – 447 с. – (Высшее образование: Бакалавриат). – ISBN 978-5-00091-458-8. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1092167> (дата обращения: 23.03.2025). – Режим доступа: по подписке.
2. Мурадханов С.Э. Разработка на языке C# приложений с графическим интерфейсом: использование Windows Forms: учебник / С.Э. Мурадханов. – Москва: Изд. Дом НИТУ «МИСиС», 2019. – 396 с. – ISBN 978-5-907061-36-1. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1232758> (дата обращения: 23.03.2025). – Режим доступа: по подписке.
3. Хорев П.Б. Объектно-ориентированное программирование с примерами на C#: учебное пособие / П.Б. Хорев. – Москва: ФОРУМ: ИНФРА-М, 2023. – 200 с. – (Среднее профессиональное образование). – ISBN 978-5-00091-713-8. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1895650> (дата обращения: 23.03.2025). – Режим доступа: по подписке.

ОЦЕНКИ УРОВНЯ АДАПТАЦИИ СТУДЕНТОВ НА ОСНОВЕ НЕЧЕТКИХ МНОЖЕСТВ

*О.Н. Фисоченко^а, к.т.н., доц., С.В. Лигачев, студент
Юргинский технологический институт (филиал)*

*Национального исследовательского Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
E-mail: ^аfisochenko@tpu.ru*

Аннотация: В данной статье предложен алгоритм, на основе метода нечетких множеств, который позволяет успешно классифицировать уровень адаптации студентов для любого набора исходных признаков.

Ключевые слова: адаптация студентов, оценка уровня адаптации, метод нечетких множеств

Abstract: This article proposes an algorithm based on the fuzzy set method, which allows you to successfully classify the level of adaptation of students for any set of initial features.

Keyword: student adaptation, adaptation level assessment, fuzzy set method

В данном исследовании для диагностики уровня адаптации студентов использован метод нечетких множеств. Объектом классификации является уровень адаптации студентов.

Для выбора вариантов, измеряемых с помощью нечетких параметров, используются различные методы – ранжирование альтернатив на множестве лингвистических оценок, многокритериальный выбор на основе нечеткого отношения предпочтения, нечеткий вывод на множестве правил-продукций и т. д.

Обобщенная оценка представлена в виде:

$$\mu_k = \left(\sum_{i=1}^n \lambda_i \mu_{i,k} \right) / n ,$$

где k – номер объекта, i – номер признака, λ_i – вес i – го признака; $\mu_{i,k}$ – значение функции принадлежности i – го признака x_i для i – го признака k – го объекта; n – количество признаков.

Функции принадлежности формируются на основе метода интервальных оценок (рис. 1).

Алгоритм построения классификационной модели включает в себя следующие этапы:

1. Определение перечня признаков $y=(y_1, \dots, y_j, \dots, y_n)$, характеризующих объект.
2. Определение диапазона значений каждого признака.
3. Задание значимости w_j каждого признака в общей оценке объекта.
4. Определение перечня значений лингвистической переменной, применяемых для оценки качества и в дальнейшем обозначаемые как классы $K=(K_1, \dots, K_s, \dots, K_m)$.
5. Задание функции принадлежности каждому классу. Выбирается форма функции принадлежности, характеризующая нечеткую неопределенность границ между соседними классами. Диапазон значений каждого признака делится на m частей – по числу классов[4].