

УДК 004

**Тляумбетов Шамиль Радиевич**, магистрант, Казанский национальный исследовательский технический университет им. А.Н.

Туполева – КАИ, г. Казань

## **АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ ПОШАГОВОГО РЕШЕНИЯ ЗАДАЧ ТЕОРИИ ВЕРОЯТНОСТЕЙ**

### **Аннотация**

Статья посвящена разработке алгоритмического обеспечения для системы пошагового решения задач теории вероятностей. Предложена классификация задач по типам алгоритмов решения, включающая пять основных категорий. Разработана структура алгоритмических предписаний, состоящая из четырёх типов шагов: идентификация, выбор формулы, вычисление и верификация. Представлен алгоритм автоматической идентификации типа задачи на основе анализа ключевых слов и паттернов условия. Описан алгоритм проверки численных ответов с учётом допустимой погрешности и альтернативных форм записи. Рассмотрена программная реализация алгоритмического обеспечения на языке TypeScript

### **Annotation**

This paper focuses on developing an algorithmic framework for a step-by-step probability problem solver. The authors propose a classification of problems into five major categories based on their underlying solution algorithms. They design a structure for algorithmic instructions that comprises four distinct stages: identification, formula selection, calculation, and verification. The paper presents an algorithm for automated problem-type identification using keyword and pattern analysis within the problem statement. Additionally, it describes a numerical answer verification algorithm that accounts for acceptable error margins and alternative formats. Finally, the practical software implementation

of this algorithmic framework in TypeScript is discussed.

**Ключевые слова:** теория вероятностей, алгоритмические предписания, классификация задач, автоматическая проверка, обучающие системы, пошаговое решение, TypeScript

**Keywords:** probability theory, algorithmic instructions, problem classification, automated verification, educational systems, step-by-step problem-solving, TypeScript.

## **Введение**

Теория вероятностей как учебная дисциплина обладает спецификой, затрудняющей её освоение традиционными методами. Задачи данного раздела математики требуют не только знания формул, но и умения строить вероятностные модели реальных ситуаций, интерпретировать полученные результаты. Многообразие типов задач и методов их решения создаёт дополнительные трудности для обучающихся, особенно на этапе выбора подходящего подхода.

Разработка алгоритмического обеспечения для системы пошагового решения задач предполагает решение нескольких взаимосвязанных проблем: классификацию задач по типам алгоритмов решения, формализацию самих алгоритмов в виде, пригодном для программной реализации, разработку методов идентификации типа задачи и проверки ответов. Целью настоящей работы является создание алгоритмического обеспечения, интегрирующего указанные компоненты в единую систему.

## **Классификация задач теории вероятностей**

Анализ типовых задач выявил пять основных категорий: классическое определение вероятности; теоремы сложения и умножения, требующие анализа зависимостей событий; формулы полной вероятности и Байеса, оперирующие системой гипотез; схема Бернулли для повторных испытаний и вычисление числовых характеристик случайных величин.

Таблица 1 - Классификация задач теории вероятностей

№	Категория	Основная формула	Ключевые признаки
1	Классическое определение	$P(A) = m/n$	Равновероятные исходы, выбор элементов
2	Теоремы сложения и умножения	$P(A \cup B), P(A \cap B)$	«Хотя бы один», «и», «или»
3	Полная вероятность, Байес	$\sum P(H_i)P(A H_i)$	Гипотезы, условия, переоценка
4	Схема Бернулли	$P_n(k) = C_k^n p^k q^{n-k}$	n испытаний, k успехов, p=const
5	Числовые характеристики	$M(X), D(X), \sigma(X)$	Случайная величина, распределение

### Структура алгоритмических предписаний

Алгоритмическое предписание представляет собой обобщённый метод решения, формализованный в виде четырёх типов шагов (табл. 2).

Таблица 2 - Структура алгоритмических предписаний

Тип шага	Назначение	Содержание действий
Идентификация	Анализ условия задачи	Выделение типа эксперимента, множества исходов, событий и искомых величин.
Формула	Выбор математической модели	Подбор формулы, определение переменных и условий применимости.
Вычисление	Получение численного результата	Выполнение арифметических операций с контролем типичных ошибок.
Верификация	Проверка корректности ответа	Оценка принадлежности значения интервалу $[0, 1]$ и логическая интерпретация.

### Программная реализация и архитектура системы

Алгоритмическое обеспечение реализовано в рамках веб-приложения «Probability Learning App», построенного на современном технологическом стеке. Использование фреймворка Next.js 16 (App Router) и языка TypeScript позволило создать типизированную, масштабируемую архитектуру с четким

разделением ответственности между серверной и клиентской частями приложения. В качестве системы управления базой данных выбрана СУБД SQLite, интегрированная через ORM-слой Prisma, что обеспечивает типобезопасность запросов и простоту развертывания.

### **Архитектура приложения**

При проектировании системы особое внимание уделено производительности и доступности учебного контента. Основные контентные страницы (главная страница, разделы теории и практики) реализованы как асинхронные серверные компоненты (Async Server Components). В отличие от традиционной клиентской отрисовки, данные из базы данных запрашиваются на сервере через Prisma, после чего передаются в клиентские компоненты-обертки (HomeContent, TheoryContent, PracticeContent) в виде сериализованных объектов.

Данный подход позволяет сформировать HTML на сервере и отправить клиенту готовый контент, что устраняет эффект смещения макета (CLS) и значительно ускоряет отрисовку крупнейшего контентного блока (LCP). Клиентские компоненты отвечают исключительно за интерактивность: навигацию, отображение прогресса и обработку действий пользователя.

### **Реализация алгоритмических предписаний**

В соответствии с разработанной теоретической моделью, структура шагов алгоритма формализована в виде TypeScript-интерфейсов. Центральным элементом является интерфейс `AlgorithmStep`, описывающий один шаг предписания.

Листинг 1 - Описание структуры шага алгоритма на языке TypeScript

```
type StepType = 'identify' | 'formula' | 'calculate'
| 'verify';

interface AlgorithmStep {
  step: number;
  title: string;
  description: string;
  type: StepType;
  formula?: string;
```

```
    hint?: string;  
  }
```

Как видно из листинга, поле `type` напрямую кодирует теоретическую классификацию шагов (идентификация, формула, вычисление, верификация). Экземпляры данных интерфейсов хранятся в базе данных в формате JSON, что позволяет редактировать алгоритмические предписания без изменения программного кода. Для визуализации математических формул, содержащихся в поле `formula`, используется библиотека KaTeX, обеспечивающая корректный рендеринг LaTeX-нотации в браузере.

### **Алгоритм проверки ответов**

Модуль проверки ответов реализован в виде API-маршрута `POST /api/problems/[id]/check`. При получении ответа пользователя система выполняет многоуровневую проверку.

На первом этапе производится нормализация входных данных: удаление пробелов, замена запятых на точки для унификации десятичных дробей. На втором этапе выполняется сравнение численного значения с эталонным ответом с заданной точностью (`tolerance = 0.001`). Это позволяет системе засчитывать верные ответы, записанные в разной форме (например, десятичную дробь  $0.25$  и обыкновенную  $1/4$ ).

Поскольку система не предусматривает обязательной регистрации, идентификация пользователей осуществляется через `sessionId`, генерируемый на клиенте и сохраняемый в `localStorage`. Результат проверки сохраняется в таблице `UserProgress` с привязкой к текущей сессии, что обеспечивает непрерывность образовательного трека.

### **Интерфейс системы**

Разработанный веб-интерфейс обеспечивает интуитивно понятное взаимодействие с системой. Обучающийся последовательно выбирает тему из списка, изучает теоретический материал с визуализированными формулами и переходит к разделу практики.

Ключевым элементом интерфейса является компонент пошагового

решения (SolutionStepper). Он визуализирует алгоритмическое предписание: пользователь видит текущий этап решения, получает текстовые подсказки и ссылки на теоретические формулы. Для ввода ответа предусмотрен интерфейс, поддерживающий различные форматы записи чисел, что снижает когнитивную нагрузку на обучающегося.

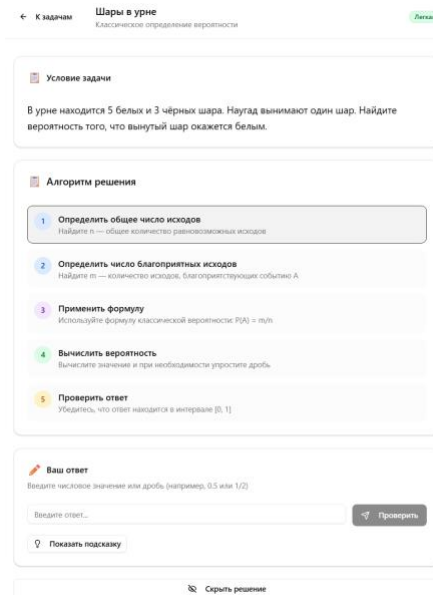


Рисунок 1. Интерфейс пошагового решения задачи

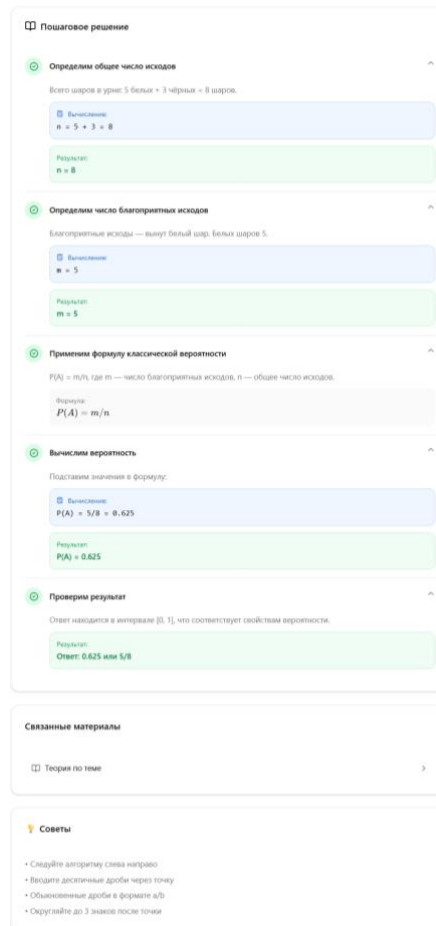


Рисунок 2. Интерфейс пошагового решения задачи

## Заключение

В результате работы разработано алгоритмическое и программное обеспечение для системы пошагового решения задач теории вероятностей.

Предложена классификация задач по пяти категориям, основанная на методе решения. Разработана структура алгоритмических предписаний, включающая четыре типа шагов: идентификация, формула, вычисление, верификация.

Программная реализация выполнена на стеке Next.js 16 и TypeScript. Применение архитектуры Server Components и Prisma ORM обеспечило высокую производительность приложения и типобезопасность данных.

Разработан модуль проверки ответов, учитывающий погрешность вычислений и альтернативные формы записи.

Система готова к практическому применению в учебном процессе и может быть расширена за счет добавления новых типов задач и методов решения.

### Список литературы

1. Ланда Л. Н. Алгоритмизация в обучении / Л. Н. Ланда. – М.: Просвещение, 2019. – 523 с.
2. Гмурман В. Е. Руководство к решению задач по теории вероятностей и математической статистике / В. Е. Гмурман. – М.: Юрайт, 2021. – 404 с.
3. Кремер Н. Ш. Теория вероятностей и математическая статистика / Н. Ш. Кремер. – М.: Юрайт, 2021. – 538 с.
4. Кормен Т. Х. Алгоритмы: построение и анализ / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – М.: Вильямс, 2021. – 1328 с.
5. Талызина Н. Ф. Управление процессом усвоения знаний / Н. Ф. Талызина. – М.: Изд-во МГУ, 2018. – 344 с.
6. Документация TypeScript [Электронный ресурс]. – URL: <https://www.typescriptlang.org/docs> (дата обращения: 20.01.2025).
7. Тюрин Ю. Н. Теория вероятностей и статистика / Ю. Н. Тюрин, А. А. Макаров, И. Р. Высоцкий, И. В. Ященко. – М.: МЦНМО, 2020. – 256 с.
8. Беспалько В. П. Слагаемые педагогической технологии / В. П. Беспалько. – М.: Педагогика, 2018. – 192 с.

### References

1. L. N. Landa, *Algorithmization in Education*. Moscow, Russia: Prosveshcheniye, 2019, 523
2. V. E. Gmurman, *Guide to Solving Problems in Probability Theory and Mathematical Statistics*. Moscow, Russia: Urayt, 2021, 404 p.
3. N. S. Kremer, *Probability Theory and Mathematical Statistics*. Moscow, Russia: Urayt, 2021, 538 p.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Moscow, Russia: Williams, 2021, 1328 p.

5. N. F. Talyzina, *Management of the Knowledge Acquisition Process*. Moscow, Russia: MSU Publishing House, 2018, 344 p.
6. *TypeScript Documentation*. Accessed: Jan. 20, 2025. [Online]. Available: <https://www.typescriptlang.org/docs>
7. Yu. N. Tyurin, A. A. Makarov, I. R. Vysotsky, and I. V. Yashchenko, *Probability Theory and Statistics*. Moscow, Russia: MCCNMO, 2020, 256 p.
8. V. P. Bepalko, *Components of Pedagogical Technology*. Moscow, Russia: Pedagogika, 2018, 192 p.

Обратите внимание, что основной текст статьи должен быть не менее 3 страниц (за исключением названия, информации об авторах, аннотации, ключевых слов, списка литературы и их перевода).