

## **Разработка интерактивной образовательной игры «Инфориум» для изучения теории множеств и отношений с применением искусственного интеллекта**

*Екимова Яна Сергеевна*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

Статья посвящена разработке интерактивной образовательной веб-игры «Инфориум» для изучения темы «Отношения объектов и их множеств» в курсе информатики 8–9 классов с применением искусственного интеллекта DeepSeek для генерации программного кода. Подробно рассматривается поэтапное создание приложения от формулировки развернутого промта с описанием жанра, механик, визуального стиля и требований к 15 SVG-пирожным до получения и адаптации готового HTML/CSS/JavaScript-кода, реализации игровой логики, модальных окон, системы подсказок, отслеживания прогресса, звездных достижений и финального экрана с результатами. В результате была разработана интерактивная образовательная игра с использованием искусственного интеллекта DeepSeek.

**Ключевые слова:** искусственный интеллект, информатика, HTML, CSS, javascript, deepseek.

## **Development of an interactive educational game "Inforium" for studying set theory and relationships using artificial intelligence**

*Ekimova Yana Sergeevna*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

The article is devoted to the development of an interactive educational web game "Inforium" for studying the topic "Relations of objects and their sets" in the computer science course of grades 8-9 using DeepSeek artificial intelligence to generate program code. The step-by-step creation of the application is considered in detail from the formulation of a detailed script describing the genre, mechanics, visual style and requirements for 15 SVG cakes to obtaining and adapting ready-made HTML/CSS/ JavaScript code, implementing game logic, modal windows, a hint system, tracking progress, stellar achievements and the final screen with the results. As a result, an interactive educational game using DeepSeek artificial intelligence was developed.

**Keywords:** artificial intelligence, computer science, HTML, CSS, javascript, deepseek.

## **1 Введение**

### **1.1 Актуальность**

Современный образовательный процесс, направленный на повышение вовлеченности и мотивации обучающихся, немислим без внедрения элементов геймификации и интерактивных технологий. Стремительная цифровизация и смена поколенческих предпочтений обуславливают потребность в простых и доступных инструментах, позволяющих педагогам и методистам самостоятельно создавать увлекательный игровой контент без глубоких технических знаний. Одним из эффективных решений в этой сфере является использование генеративных искусственных нейросетей, таких как DeepSeek, которые способны по текстовому запросу создавать полностью рабочий программный код интерактивных приложений, включая HTML-верстку, CSS-стилизацию, SVG-графику и JavaScript-логику.

Освоение и методическое описание подобных инструментов представляют собой значимый вклад в развитие практик смешанного и дистанционного обучения, где интерактивность и соревновательный элемент напрямую влияют на усвоение материала. Исследование и популяризация подходов к использованию искусственного интеллекта для генерации образовательных игр способствуют развитию цифровой грамотности педагогов и отвечают на вызовы времени, связанные с необходимостью быстрой адаптации учебных материалов под требования современных школьников, снижением барьеров для входа в сферу образовательного геймдизайна и массовым внедрением персонализированных цифровых инструментов в школьную практику.

### **1.2 Обзор исследований**

В статье С.В. Рогатина, В.А. Антошкина рассматривается методика разработки компьютерных игр по технологии HTML5 с помощью CASE-средства GameMakerStudio компании YoYoGamesLtd. Проводится тестирование производительности работы компьютерных игр в браузере в зависимости от сложности использованных спрайтов игровых объектов [1]. Разработали игровое приложение «league of football heads» с использованием html5, css3, javascript И.А. Фёдоров, Е.Ю. Кузьменкова [2]. Я. А. Королук, Н.Ю. Куликова показали опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайн-обучения школьников информатике [3]. В работе рассмотрели возможности языка разметки гипертекста HTML5 для создания макета электронного учебника Д.М. Размологов, А.С. Панина, П.В. Захаров. Привели разработанный макет, а также часть таблицы стилей для дизайна макета [4]. И.С. Шегаев, А.Г. Полозов разработали новейший политологический глоссарий: основные категории, понятия, термины [5].

### **1.3 Цель исследования**

Целью исследования является разработка и описание методического подхода к созданию интерактивной образовательной HTML-игры

«Инфориум» для изучения темы «Отношения объектов и их множеств» с использованием генеративного искусственного интеллекта DeepSeek.

## 2 Материалы и методы

Для разработки и тестирования интерактивной образовательной игры «Инфориум» в рамках данного исследования использовался генеративный искусственный интеллект DeepSeek (доступный по адресу <https://chat.deepseek.com>) нейросетевая языковая модель, способная создавать программный код на HTML, CSS и JavaScript на основе текстовых описаний на естественном языке. В качестве среды выполнения и тестирования использовался веб-браузер с поддержкой современных стандартов HTML5, CSS3 и ECMAScript 6, что обеспечивало корректное отображение SVG-графики, модальных окон, анимаций и конфетти-эффектов.

## 3 Результаты

Для того чтобы создать игру, необходимо пройти процедуру регистрации в искусственном интеллекте DeepSeek <https://chat.deepseek.com>, после чего сформулировать и отправить промт — текстовый запрос на естественном языке, содержащий описание жанра, механик, целевой платформы и желаемого визуального стиля.

Создай интерактивную образовательную HTML-игру [Инфориум] для предмета [Информатика], ученики [8-9 класс].

### 📖 ПАРАМЕТРЫ ИГРЫ:

- Тема заданий: [Отношения объектов и их множеств]
- Количество заданий: [15]
- Формат вопросов: {{ТИП\_ВОПРОСОВ: множественный выбор / короткий ответ / перетаскивание}}
- Сложность: {{Разный}}

### 😊 ДИЗАЙН:

- Фоновое изображение или цвет: {{ФОН: URL или описание}}
- Цветовая палитра: {{ЦВЕТА: например, "пастельные тона, синий и жёлтый"}}
- Стил графики: {{СТИЛЬ: "милые иконки", "пиксель-арт", "минимализм" и т.д.}}
- Шрифты: подключить через Google Fonts (заголовки — игровой, текст — читаемый)

### 🥟 SVG-ПИРОЖНЫЕ (отрисовать кодом, 15 штук):

Создай массив SVG-строк для кликабельных элементов. Каждое пирожное — компактный inline-SVG (viewBox="0 0 100 100"), милый стиль, яркие цвета.

#### 1. Капкейк с вишенкой:

- Бумажная формочка-трапеция (коричневая/розовая)
- Шапочка крема (овал, светлый)
- Вишенка сверху (красный круг + зелёный хвостик)
- Блик-кружочки на креме

#### 2. Макарун:

- Два овальных "крышечки" (пастельный цвет)
- Прослойка крема между ними
- Лёгкие блики, градиентная заливка

#### 3. Пончик с посыпкой:

- Кольцо (основа — коричневая/бежевая)
- Глазурь сверху (розовая/шоколадная)

- Разноцветные кружочки-посыпка
- 4. Кусочек торта:
  - Треугольный сектор
  - Слои: бисквит + крем
  - Ягода/вишенка сверху + свечка (опционально)
- 5. Круассан:
  - Изогнутая форма полумесяца
  - Золотисто-коричневый градиент
  - Линии-слои теста
- 6. Эклер:
  - Продолговатая форма
  - Шоколадная глазурь сверху
  - Контурные линии для объёма
- 7. Печенье с шоколадной крошкой:
  - Круглое золотистое печенье
  - Тёмные кружочки-капли шоколада
  - Неровные края для реалистичности
- 8. Маффин:
  - Формочка-трапеция снизу
  - Шапочка с "кратером" сверху
  - Шоколадные капли/посыпка
- 9. Тарт:
  - Песочная корзинка (эллипс)
  - Заварной крем (жёлтый градиент)
  - Ягоды по кругу
- 10. Брауни:
  - Квадратный кусочек
  - Тёмно-шоколадная текстура
  - Глянцевая корочка + крошки
- 11. Чизкейк:
  - Треугольный сектор
  - Кремовая начинка + песочная основа
  - Ягодный топинг + листик мяты
- 12. Слоеная выпечка:
  - Прямоугольник с видимыми слоями
  - Золотистая корочка
  - Линии-разделения слоёв
- 13. Фруктовый тарт:
  - Песочная основа
  - Крем + разноцветные фрукты (клубника, киви, черника)
  - Глянцевый эффект глазури
- 14. Шоколадный торт:
  - Многослойный прямоугольник
  - Шоколадная глазурь с подтёками
  - Украшение: ягода/орех/золотая крошка
- 15. Заварное пирожное:
  - Круглое заварное тесто
  - Видимая прослойка крема
  - Сахарная пудра (белые точки-блики)

## 🕒 ТЕХНИКА ОТРИСОВКИ:

- Используйте `<ellipse>`, `<path>`, `<circle>`, `<rect>` для форм
- Градиенты через `<linearGradient>` / `<radialGradient>` (опционально)
- Блик-эффекты: маленькие белые круги с `opacity`
- Все SVG должны быть ~100×100px, масштабироваться через CSS
- Храни SVG как строки в массиве: `const pastriesSVG = [ `<svg>...`, ... ]`

## 🌀 СТРУКТУРА ИГРЫ:

## ▶ СТАРТОВЫЙ ЭКРАН:

- Название игры крупно
- Краткие правила (3-5 пунктов)
- Кнопка "Начать"

## ▶ ИГРОВОЙ ЭКРАН:

- Сетка из 15 кликабельных карточек с SVG-пирожными
- При клике открывается модальное окно с:
  - Текст задания
  - 4 варианта ответа (один верный)
  - Кнопка "🕒 Подсказка" (показывает текст подсказки)
  - Авто-проверка при выборе ответа
- Визуальная обратная связь: ☒ зелёная рамка для верного, ☐ красная для неверного
- Ответенные пирожные помечаются цветом и блокируются

## ▶ ЭКРАН РЕЗУЛЬТАТОВ:

- Количество правильных / неправильных ответов
- Процент успеха
- Мотивирующее сообщение в зависимости от результата
- Кнопка "Играть снова"

## ⚙️ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ:

- Один HTML-файл (всё внутри: HTML + CSS + JS)
- Адаптивная вёрстка (мобильные + десктоп)
- Плавные анимации: hover-эффекты на пирожных, modal pop-in, конфетти при победе
- Все 15 SVG отрисованы кодом (не внешние картинки)
- Без внешних зависимостей, кроме шрифтов Google Fonts
- Комментарии в коде для учителя (где менять вопросы)

## 📄 ДАННЫЕ ЗАДАНИЙ (массив в JS):

```
[
  {
    id: 1,
    text: "{{ТЕКСТ_ВОПРОСА_1}}",
    answers: ["{{ВАРИАНТ_1}}", "{{ВАРИАНТ_2}}", "{{ВАРИАНТ_3}}", "{{ВАРИАНТ_4}}"],
    correct: {{ИНДЕКС_ВЕРНОГО_ОТВЕТА_0-3}},
    hint: "{{ТЕКСТ_ПОДСКАЗКИ}}"
  },
  // ... повторить для всех заданий
]
```

## 🌟 ДОПОЛНИТЕЛЬНО (по желанию):

- Звуковые эффекты: правильный/неправильный ответ (через Web Audio API)
- Система звёзд: +1 звезда за 5 правильных подряд
- Таймер на всё прохождение (опционально)
- Прогресс-бар: "Задание 3 из 15"
- Конфетти-анимация при 100% результате

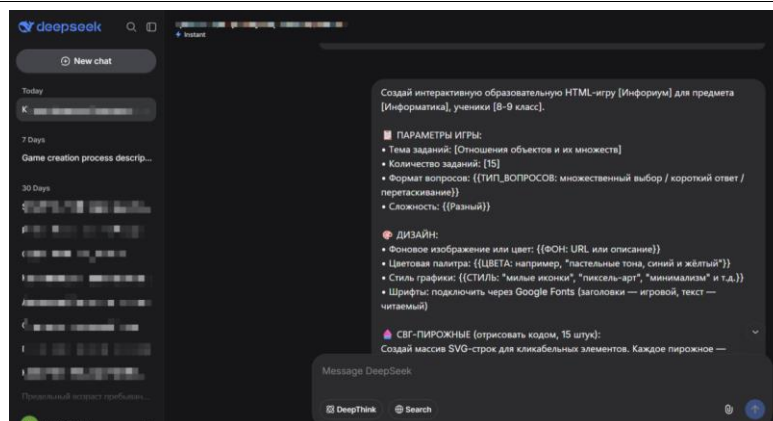


Рисунок 1. Отправка промта

Далее копируем написанный программный код на языке JavaScript с использованием HTML и CSS для реализации игровой логики, пользовательского интерфейса и управления состояниями экранов (стартовый, игровой, результативный), а также организацию обработки событий, рендеринг игровых элементов и смену визуальных состояний через манипуляции с CSS-классами.

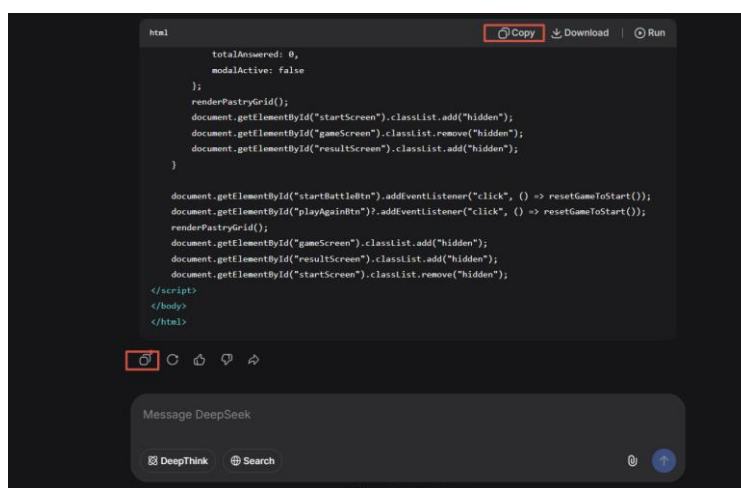


Рисунок 2. Копирование кода

Затем создаем документ и называем его index.html и вставляем туда полученный код, который представляет собой структурированную HTML-разметку с интегрированными JavaScript-функциями, обеспечивающими базовую логику работы игры, включая управление состоянием приложения, обработку пользовательского ввода и динамическое обновление содержимого игрового интерфейса.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">
  <title>Инфоориум | Отношения объектов и множеств</title>
  <!-- Google Fonts: игровой заголовок + читабельный текст -->
```

```
<link
href="https://fonts.googleapis.com/css2?family=Inter:opsz,wght@14..32,400;14..32,500;14..32,600;14..32,700&family=Orbitron:wght@500;700;900&display=swap" rel="stylesheet">
<style>
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  min-height: 100vh;
  /* СВЕТЛЫЙ ФОН: тёплый градиент для отличной читаемости */
  background: linear-gradient(145deg, #e9f0ff 0%, #d9e4ff 100%);
  font-family: 'Inter', sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 24px;
}

/* основной контейнер — светлый, с мягкой тенью */
.game-wrapper {
  max-width: 1400px;
  width: 100%;
  background: rgba(255, 255, 255, 0.92);
  backdrop-filter: blur(2px);
  border-radius: 64px;
  border: 1px solid rgba(52, 102, 182, 0.25);
  box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1), 0 0 0 1px rgba(70, 120, 200, 0.2);
  padding: 20px 28px 35px;
  transition: all 0.2s;
}

/* экраны */
.screen {
  transition: 0.3s ease;
}
.hidden {
  display: none;
}

/* СТАРТОВЫЙ ЭКРАН — читаемый */
.start-screen {
  text-align: center;
  padding: 20px 10px;
}

.game-title {
  font-family: 'Orbitron', monospace;
  font-size: 3.8rem;
  background: linear-gradient(135deg, #1a3e8c, #2b5bb3, #0f2d66);
  -webkit-background-clip: text;
  background-clip: text;
  color: transparent;
  text-shadow: 2px 2px 12px rgba(0,0,0,0.1);
  letter-spacing: 2px;
  margin-bottom: 12px;
}

.sub {
  font-size: 1.3rem;
  color: #1a3c6e;
  font-weight: 600;
  margin-bottom: 20px;
}
```

```
}
.rules-panel {
  background: #ffffffdd;
  border-radius: 48px;
  padding: 22px 30px;
  max-width: 650px;
  margin: 20px auto;
  backdrop-filter: blur(4px);
  border: 1px solid #2f6bb0;
  text-align: left;
  color: #0d2b4e;
  box-shadow: 0 6px 14px rgba(0,0,0,0.05);
}
.rules-panel h3 {
  color: #134b8e;
  font-weight: 800;
  margin-bottom: 15px;
  font-size: 1.8rem;
}
.rules-panel li {
  margin-left: 28px;
  margin-bottom: 12px;
  font-weight: 500;
  line-height: 1.4;
  color: #0f2c48;
}
}
.btn-start {
  background: linear-gradient(95deg, #2b6eff, #9f4eff);
  border: none;
  font-family: 'Orbitron', monospace;
  color: white;
  font-size: 1.7rem;
  font-weight: bold;
  padding: 14px 40px;
  border-radius: 60px;
  cursor: pointer;
  transition: 0.2s;
  box-shadow: 0 10px 0 #102a5e;
  margin-top: 20px;
}
.btn-start:active {
  transform: translateY(3px);
  box-shadow: 0 5px 0 #102a5e;
}
}

/* ИГРОВОЙ ЭКРАН — светлая карточка */
.game-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  flex-wrap: wrap;
  background: #eef3ff;
  padding: 10px 20px;
  border-radius: 60px;
  margin-bottom: 25px;
  gap: 15px;
  border: 1px solid #bed3ff;
}
}
.progress-badge {
  background: #ffffff;
  padding: 6px 20px;
  border-radius: 40px;
  font-weight: 700;
```

```
    color: #0e3a77;
    font-size: 1rem;
    box-shadow: 0 2px 5px rgba(0,0,0,0.05);
  }
  .star-rating {
    display: flex;
    gap: 10px;
    font-size: 1.9rem;
    background: #ffffff;
    padding: 5px 18px;
    border-radius: 50px;
  }
  .pastries-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(115px, 135px));
    justify-content: center;
    gap: 24px;
    margin: 30px 0;
  }
  .pastry-card {
    background: #ffffffea;
    border-radius: 38px;
    padding: 15px 5px 8px;
    cursor: pointer;
    transition: all 0.2s ease;
    border: 1px solid #b9d0ff;
    box-shadow: 0 8px 14px rgba(0, 0, 0, 0.08);
    display: flex;
    flex-direction: column;
    align-items: center;
  }
  .pastry-card:hover {
    transform: translateY(-8px);
    background: #ffffff;
    border-color: #3e7ed3;
    box-shadow: 0 16px 24px rgba(0, 68, 128, 0.15);
  }
  .pastry-card.answered-correct {
    background: #d4ffdd;
    border-color: #2e9e4a;
    box-shadow: 0 0 8px #7fda9a;
    cursor: default;
  }
  .pastry-card.answered-wrong {
    background: #ffe0db;
    border-color: #e6705a;
    cursor: default;
  }
  .pastry-svg {
    width: 85px;
    height: 85px;
    margin-bottom: 8px;
    pointer-events: none;
    filter: drop-shadow(0 2px 4px rgba(0,0,0,0.1));
  }
  .card-label {
    font-size: 0.7rem;
    font-weight: 700;
    background: #eef3fc;
    padding: 4px 12px;
    border-radius: 30px;
    color: #1a497c;
  }
```

```
/* Модальное окно — светлое, контрастное */
.modal {
  position: fixed;
  top: 0; left: 0;
  width: 100%; height: 100%;
  background: rgba(0, 0, 0, 0.5);
  backdrop-filter: blur(5px);
  display: flex;
  align-items: center;
  justify-content: center;
  z-index: 1100;
}
.modal-card {
  background: #ffffff;
  border-radius: 64px;
  max-width: 550px;
  width: 90%;
  padding: 28px 28px 35px;
  border: 1px solid #7fa9e0;
  box-shadow: 0 20px 40px rgba(0,0,0,0.2);
  animation: modalPop 0.25s cubic-bezier(0.2, 0.9, 0.4, 1.1);
}
@keyframes modalPop {
  from { opacity: 0; transform: scale(0.92);}
  to { opacity: 1; transform: scale(1);}
}
.question-text {
  font-size: 1.6rem;
  font-weight: 700;
  color: #0a2a4e;
  margin: 12px 0;
  text-align: center;
}
}
.answers-grid {
  display: flex;
  flex-direction: column;
  gap: 14px;
  margin: 25px 0;
}
}
.answer-option {
  background: #f1f6fe;
  border: 1px solid #628dce;
  border-radius: 100px;
  padding: 12px 20px;
  font-weight: 600;
  font-size: 1rem;
  color: #0b2b4f;
  text-align: left;
  cursor: pointer;
  transition: 0.1s;
}
}
.answer-option:hover {
  background: #e2ecff;
  border-color: #2f6bc2;
}
}
.hint-button {
  background: #fef2db;
  border: 1px solid #e0b87a;
  padding: 8px 20px;
  border-radius: 40px;
  font-weight: bold;
  color: #a5571e;
}
```

```
        margin-top: 5px;
        cursor: pointer;
    }
    .feedback-message {
        margin-top: 12px;
        font-weight: bold;
        font-size: 1rem;
    }
    .modal-close-btn {
        background: #cbdffc;
        border: none;
        margin-top: 20px;
        padding: 8px 28px;
        border-radius: 40px;
        color: #1f4a81;
        font-weight: bold;
        cursor: pointer;
    }
    .correct-feedback { color: #1f7832; }
    .wrong-feedback { color: #d4523a; }

/* Результаты */
.result-screen {
    text-align: center;
    padding: 20px;
}
.result-stats {
    background: #ebf2ff;
    padding: 25px;
    border-radius: 60px;
    font-size: 1.9rem;
    margin: 25px auto;
    color: #0e3c72;
    font-weight: bold;
}
.motivation-message {
    font-size: 1.3rem;
    margin: 15px;
    color: #1d4e89;
}
.btn-restart {
    background: linear-gradient(95deg, #ff9a4a, #ff6a3d);
    border: none;
    padding: 14px 38px;
    font-size: 1.5rem;
    border-radius: 60px;
    font-weight: bold;
    color: white;
    cursor: pointer;
    margin-top: 15px;
}
canvas.confetti-canvas {
    position: fixed;
    top: 0;
    left: 0;
    pointer-events: none;
    z-index: 2000;
}
footer {
    text-align: center;
    font-size: 0.75rem;
    color: #5b7fb0;
    margin-top: 20px;
```

```

    }
    @media (max-width: 650px) {
        .game-title { font-size: 2.4rem; }
        .pastries-grid { gap: 14px; }
    }
</style>
</head>
<body>
<div class="game-wrapper" id="infoGameApp">
    <!-- СТАРТОВЫЙ ЭКРАН -->
    <div id="startScreen" class="screen start-screen">
        <div class="game-title">⚡ ИНФОРИУМ ⚡</div>
        <div class="sub">Отношения объектов и их множеств</div>
        <div class="rules-panel">
            <h3>📋 ПРОТОКОЛ ЗАДАНИЙ</h3>
            <ul>
                <li>💎 15 пирожных-вопросов по теме «Множества и отношения».</li>
                <li>💎 Кликни на любое — откроется миссия с 4 вариантами.</li>
                <li>💎 Выбери верный ответ, получи обратную связь.</li>
                <li>💎 За 5 правильных подряд — звезда (макс 5 ☆).</li>
                <li>💎 Все отвеченные пирожные блокируются. Добейся 100%!</li>
            </ul>
        </div>
        <button class="btn-start" id="startBattleBtn">🚀 ЗАПУСТИТЬ СЕАНС</button>
    </div>

    <!-- ИГРОВОЙ ЭКРАН -->
    <div id="gameScreen" class="screen hidden">
        <div class="game-header">
            <div class="progress-badge" id="progressCounter">Заданий пройдено: 0 / 15</div>
            <div class="star-rating" id="starRatingDisplay">☆☆☆☆☆</div>
        </div>
        <div class="pastries-grid" id="pastriesGrid"></div>
        <footer>🧁 Каждое пирожное — новая задача. Успехов в анализе множеств!</footer>
    </div>

    <!-- ЭКРАН РЕЗУЛЬТАТОВ -->
    <div id="resultScreen" class="screen hidden">
        <div class="result-screen">
            <div class="game-title">📊 ОТЧЕТ</div>
            <div class="result-stats" id="finalStats"></div>
            <div id="resultMotivation" class="motivation-message"></div>
            <button class="btn-restart" id="playAgainBtn">🎮 ИГРАТЬ СНОВА</button>
        </div>
    </div>
</div>

<script>
    // ----- 15 SVG-пирожных (милые / яркие, inline) -----
    const pastriesSVG = [
        `<svg viewBox="0 0 100 100"><path d="M32,64 L68,64 L62,85 L38,85 Z" fill="#E8B0A0"/><ellipse
        cx="50" cy="50" rx="24" ry="20" fill="#FFF2D7"/><circle cx="50" cy="28" r="7" fill="#E34234"/><path
        d="M50,22 L53,17 L47,17Z" fill="#6B8E23"/><circle cx="40" cy="47" r="2.5" fill="white" opacity="0.8"/><circle
        cx="60" cy="45" r="2" fill="white"/></svg>`,
        `<svg viewBox="0 0 100 100"><ellipse cx="50" cy="54" rx="25" ry="32" fill="#FFB7D2"/><ellipse
        cx="50" cy="50" rx="24" ry="30" fill="#FF9FC0"/><ellipse cx="50" cy="58" rx="15" ry="6"
        fill="#FFF4E6"/><circle cx="36" cy="44" r="2.5" fill="white"/><circle cx="64" cy="46" r="2"
        fill="white"/></svg>`,
        `<svg viewBox="0 0 100 100"><path d="M25,45 A25,25 0 0,1 75,45 L75,70 A25,25 0 0,1 25,70 Z"
        fill="#E8B886"/><ellipse cx="50" cy="43" rx="32" ry="14" fill="#FFADAD"/><circle cx="33" cy="43" r="3"

```

```

fill="#FADB67"/><circle cx="50" cy="41" r="3" fill="#AAD6FF"/><circle cx="67" cy="43" r="3"
fill="#6FCF97"/></svg>`,
    <svg viewBox="0 0 100 100"><polygon points="52,18 82,74 18,74" fill="#E7B67A"/><rect x="35"
y="70" width="30" height="9" fill="#A26232"/><rect x="30" y="62" width="40" height="10"
fill="#FDEBD2"/><circle cx="52" cy="32" r="5" fill="#E34234"/><rect x="50" y="22" width="4" height="12"
fill="#E3B87C"/></svg>`,
    <svg viewBox="0 0 100 100"><path d="M28,58 C28,38 72,38 72,58 C72,75 28,75 28,58"
fill="url(#croissantGrad)"/><defs><linearGradient id="croissantGrad"><stop offset="0%"
stop-color="#F3CF7A"/><stop offset="100%" stop-color="#CA984A"/></linearGradient></defs><path d="M32,52
Q50,46 68,55" stroke="#B57236" fill="none" stroke-width="2"/></svg>`,
    <svg viewBox="0 0 100 100"><rect x="28" y="48" width="44" height="34" rx="10"
fill="#DEB887"/><rect x="25" y="42" width="50" height="12" rx="6" fill="#573E1E"/><circle cx="50" cy="46"
r="2.5" fill="white" opacity="0.6"/></svg>`,
    <svg viewBox="0 0 100 100"><circle cx="50" cy="56" r="33" fill="#F5D298"/><circle cx="35"
cy="47" r="5" fill="#553B1C"/><circle cx="58" cy="50" r="5.5" fill="#553B1C"/><circle cx="45" cy="59" r="4.5"
fill="#553B1C"/></svg>`,
    <svg viewBox="0 0 100 100"><path d="M30,72 L70,72 L64,42 L36,42 Z" fill="#BE7C4A"/><ellipse
cx="50" cy="40" rx="24" ry="16" fill="#FFE0BA"/><circle cx="40" cy="38" r="3" fill="#4E2C14"/><circle
cx="60" cy="36" r="3" fill="#4E2C14"/></svg>`,
    <svg viewBox="0 0 100 100"><ellipse cx="50" cy="62" rx="34" ry="20" fill="#EAC394"/><ellipse
cx="50" cy="56" rx="28" ry="12" fill="#FEF3CA"/><circle cx="30" cy="54" r="4.5" fill="#FF6B6B"/><circle
cx="50" cy="51" r="4.5" fill="#99D17B"/><circle cx="70" cy="54" r="4.5" fill="#FFA559"/></svg>`,
    <svg viewBox="0 0 100 100"><rect x="28" y="48" width="44" height="32" rx="6"
fill="#3E2413"/><rect x="28" y="44" width="44" height="10" rx="4" fill="#5E3318"/><circle cx="42" cy="60"
r="2.5" fill="#996F48"/><circle cx="58" cy="62" r="2.5" fill="#996F48"/></svg>`,
    <svg viewBox="0 0 100 100"><polygon points="50,24 80,72 20,72" fill="#FFF5E0"/><rect x="32"
y="68" width="36" height="8" fill="#D9AE7C"/><ellipse cx="50" cy="56" rx="12" ry="6" fill="#D94343"/><path
d="M64,47 L72,44 L68,53 Z" fill="#598B2F"/></svg>`,
    <svg viewBox="0 0 100 100"><rect x="26" y="46" width="48" height="36" rx="4"
fill="#EDC282"/><line x1="28" y1="55" x2="72" y2="55" stroke="#BD8A4A" stroke-width="1.5"/><line x1="28"
y1="63" x2="72" y2="63" stroke="#BD8A4A" stroke-width="1.5"/></svg>`,
    <svg viewBox="0 0 100 100"><ellipse cx="50" cy="65" rx="38" ry="22" fill="#E6BB78"/><ellipse
cx="50" cy="58" rx="30" ry="14" fill="#FEF4DE"/><circle cx="32" cy="52" r="5" fill="#FF625D"/><circle
cx="50" cy="48" r="5" fill="#48C76B"/><circle cx="68" cy="52" r="5" fill="#3E9ED4"/></svg>`,
    <svg viewBox="0 0 100 100"><rect x="28" y="38" width="44" height="44" rx="8"
fill="#4D2F17"/><rect x="26" y="33" width="48" height="12" fill="#A5551F"/><circle cx="50" cy="28" r="5"
fill="#E7BC85"/><circle cx="62" cy="38" r="3" fill="#DFA753"/></svg>`,
    <svg viewBox="0 0 100 100"><circle cx="50" cy="57" r="28" fill="#FADCAD"/><ellipse cx="50"
cy="62" rx="18" ry="10" fill="#FFF9E8"/><circle cx="40" cy="49" r="3" fill="white" opacity="0.8"/><circle
cx="60" cy="49" r="3" fill="white"/></svg>`
];

```

// ----- ВОПРОСЫ по теме "Отношения объектов и их множеств" (8-9 класс) -----

```

const questionsSet = [
    { text: "Какое отношение описывает фразу: «Каждый ученик учится в некотором классе?»",
      answers: ["один-ко-многим", "многие-ко-многим", "один-к-одному", "пустое отношение"], correct: 0, hint:
      "Один ученик принадлежит одному классу, но класс содержит много учеников." },
    { text: "Пусть  $A = \{2,4,6,8\}$ ,  $B = \{1,2,3,4\}$ . Отношение «число  $x$  из  $A$  больше числа  $y$  из  $B$ ».
      Сколько пар в этом отношении?", answers: ["6", "8", "10", "12"], correct: 1, hint: "Перебери пары, где элемент
       $A >$  элемента  $B$ ." },
    { text: "Какое из множеств является подмножеством другого?  $A = \{ \text{кошки, собаки} \}$ ,  $B = \{ \text{животные} \}$ ",
      answers: [" $A \subset B$ ", " $B \subset A$ ", " $A = B$ ", "не сравнимы"], correct: 0, hint: "Все кошки и собаки —
      животные." },
    { text: "Отношение «быть делителем» на множестве  $\{2,4,6,8\}$  является:", answers:
      ["рефлексивным", "симметричным", "транзитивным и рефлексивным", "антисимметричным"], correct: 2, hint:
      "Число делит само себя, и если  $a|b$  и  $b|c$ , то  $a|c$ ." },
    { text: "Какое из утверждений описывает отношение « $x$  имеет общий цвет с  $y$ » для множества
      машин?", answers: ["симметричное", "транзитивное", "рефлексивное и симметричное", "антисимметричное"],
      correct: 2, hint: "Любая машина имеет общий цвет с собой, и если  $x$  с  $y$ , то  $y$  с  $x$ ." },
    { text: "Дано множество студентов. Отношение «учиться в одной группе» — это:", answers:
      ["эквивалентности", "порядка", "функциональное", "толерантности"], correct: 0, hint: "Оно рефлексивно,
      симметрично и транзитивно." },

```

```

    { text: "Пусть  $R = \{(1,2), (2,3), (1,3)\}$ . Какое свойство выполняется?", answers: ["транзитивность", "симметричность", "рефлексивность", "антисимметричность"], correct: 0, hint: "Есть (1,2) и (2,3) и (1,3) — транзитивно." },
    { text: "Отношение «являться отцом» на множестве людей:", answers: ["антисимметрично", "симметрично", "рефлексивно", "транзитивно"], correct: 0, hint: "Если А отец В, то В не может быть отцом А." },
    { text: "Сколько элементов содержит декартово произведение  $A \times B$ , если  $|A|=4$ ,  $|B|=3$ ?", answers: ["12", "7", "8", "16"], correct: 0, hint: "Мощность произведения =  $|A| \cdot |B|$ ." },
    { text: "Какое отношение задано графом, где каждая вершина соединена сама с собой и нет других рёбер?", answers: ["тождество", "универсальное", "пустое", "порядок"], correct: 0, hint: "Только пары (x,x)." },
    { text: "Что означает запись  $R \subseteq A \times B$ ?", answers: ["R — отношение из A в B", "R — подмножество A", "B — область значений", "кортеж"], correct: 0, hint: "Бинарное отношение — подмножество прямого произведения." },
    { text: "Для отношения « $x \leq y$ » на натуральных числах верно:", answers: ["антисимметрично и транзитивно", "симметрично", "рефлексивно и симметрично", "только рефлексивно"], correct: 0, hint: "Если  $a \leq b$  и  $b \leq a \Rightarrow a=b$  и транзитивность." },
    { text: "Отношение «иметь одинаковый остаток при делении на 3» — это:", answers: ["классы эквивалентности", "частичный порядок", "функция", "инъекция"], correct: 0, hint: "Разбивает числа на 3 класса эквивалентности." },
    { text: "Даны  $A = \{1,2,3\}$ . Какой вид имеет отношение  $R = \{(1,2), (2,1), (1,1)\}$ ?", answers: ["симметричное", "антисимметричное", "транзитивное", "рефлексивное"], correct: 0, hint: "Если (1,2) то есть (2,1); симметрия есть." },
    { text: "Пусть R — «знать язык программирования». Какое из свойств необязательно?", answers: ["транзитивность", "рефлексивность", "симметричность", "антисимметричность"], correct: 0, hint: "Если А знает Python, а В знает Python, это не значит, что А знает язык C." }
  ];

```

```

// состояние игры
let game = {
  answered: new Array(15).fill(false),
  correctMap: new Array(15).fill(false),
  consecutiveCorrect: 0,
  stars: 0,
  totalAnswered: 0,
  modalActive: false
};

function updateUIStats() {
  const answeredCount = game.answered.filter(v => v === true).length;
  document.getElementById("progressCounter").innerHTML = `Заданий пройдено: ${answeredCount} / 15`;

  let starStr = "";
  for (let i = 0; i < 5; i++) starStr += (i < game.stars) ? "☆" : "☆";
  document.getElementById("starRatingDisplay").innerHTML = starStr;
}

function renderPastryGrid() {
  const grid = document.getElementById("pastriesGrid");
  if (!grid) return;
  grid.innerHTML = "";
  for (let i = 0; i < pastriesSVG.length; i++) {
    const card = document.createElement("div");
    card.className = "pastry-card";
    if (game.answered[i]) {
      if (game.correctMap[i]) card.classList.add("answered-correct");
      else card.classList.add("answered-wrong");
    }
    card.setAttribute("data-idx", i);
    const svgDiv = document.createElement("div");
    svgDiv.className = "pastry-svg";
    svgDiv.innerHTML = pastriesSVG[i];
    const labelSpan = document.createElement("span");

```

```

        labelSpan.className = "card-label";
        labelSpan.innerText = game.answered[i] ? (game.correctMap[i] ? "✓ решено" : "✗ ошибка") : "❓";
    задание";

    card.appendChild(svgDiv);
    card.appendChild(labelSpan);
    card.addEventListener("click", (e) => {
        const idx = parseInt(card.getAttribute("data-idx"));
        if (!game.answered[idx] && !game.modalActive) openQuestionModal(idx);
        else if (game.answered[idx]) showToast("Ты уже отвечал на это пирожное!", "#f8b88b");
    });
    grid.appendChild(card);
}
updateUIStats();
}

function showToast(msg, bg) {
    const div = document.createElement("div");
    div.innerText = msg;
    div.style.position = "fixed";
    div.style.bottom = "30px";
    div.style.left = "50%";
    div.style.transform = "translateX(-50%)";
    div.style.backgroundColor = bg;
    div.style.color = "#1f3b62";
    div.style.fontWeight = "bold";
    div.style.padding = "10px 24px";
    div.style.borderRadius = "40px";
    div.style.zIndex = "3000";
    div.style.backdropFilter = "blur(8px)";
    document.body.appendChild(div);
    setTimeout(() => div.remove(), 1500);
}

function openQuestionModal(id) {
    if (game.modalActive) return;
    game.modalActive = true;
    const q = questionsSet[id];
    if (!q) return;
    const modalDiv = document.createElement("div");
    modalDiv.className = "modal";
    modalDiv.innerHTML = `
        <div class="modal-card">
            <div class="question-text">❓ ${q.text}</div>
            <div class="answers-grid" id="modalAnswers"></div>
            <button class="hint-button" id="modalHintBtn">💡 Показать подсказку</button>
            <div id="modalHintMsg" style="font-size:0.85rem; color:#205493; margin:8px 0;"></div>
            <div id="modalFeedback" class="feedback-message"></div>
            <button class="modal-close-btn" id="closeModalBtn">Закрыть</button>
        </div>
    `;
    document.body.appendChild(modalDiv);
    const answersContainer = modalDiv.querySelector("#modalAnswers");
    q.answers.forEach((ans, idx) => {
        const btn = document.createElement("button");
        btn.className = "answer-option";
        btn.innerText = `${String.fromCharCode(65+idx)}. ${ans}`;
        btn.addEventListener("click", () => handleAnswer(id, idx, modalDiv, q));
        answersContainer.appendChild(btn);
    });
    modalDiv.querySelector("#modalHintBtn").addEventListener("click", () => {
        modalDiv.querySelector("#modalHintMsg").innerText = `💡 ${q.hint}`;
    });
}

```

```

modalDiv.querySelector("#closeModalBtn").addEventListener("click", () => {
  modalDiv.remove();
  game.modalActive = false;
});
}

function handleAnswer(questId, selectedIdx, modalElem, questionObj) {
  if (game.answered[questId]) return;
  const isCorrect = (selectedIdx === questionObj.correct);
  const feedbackDiv = modalElem.querySelector("#modalFeedback");
  if (isCorrect) {
    feedbackDiv.innerHTML = "✅ Превосходно! Ответ верный.";
    feedbackDiv.className = "feedback-message correct-feedback";
    game.correctMap[questId] = true;
    game.consecutiveCorrect++;
    if (game.consecutiveCorrect === 5 && game.stars < 5) {
      game.stars = Math.min(5, game.stars + 1);
      showToast("🌟 ЗВЕЗДА ПОЛУЧЕНА! 5 верных подряд!", "#f2b45a");
      if (game.stars === 5) launchConfetti();
    }
  } else {
    feedbackDiv.innerHTML = `❌ Неверно. Верный ответ: ${questionObj.answers[questionObj.correct]}`;
    feedbackDiv.className = "feedback-message wrong-feedback";
    game.correctMap[questId] = false;
    game.consecutiveCorrect = 0;
  }
  game.answered[questId] = true;
  game.totalAnswered = game.answered.filter(v=>v).length;
  renderPastryGrid();
  const btns = modalElem.querySelectorAll(".answer-option");
  btns.forEach(btn => btn.disabled = true);
  setTimeout(() => {
    if (modalElem && modalElem.parentNode) modalElem.remove();
    game.modalActive = false;
    if (game.totalAnswered === 15) showFinalScreen();
  }, 1300);
}

function showFinalScreen() {
  const correctCount = game.correctMap.filter(v=>v===true).length;
  const percent = Math.round((correctCount/15)*100);
  document.getElementById("finalStats").innerHTML = `🎯 Результат: ${correctCount} / 15 (${percent}%)`;
  let msg = "";
  if (percent === 100) msg = "🌟 АБСОЛЮТНЫЙ УРОВЕНЬ! Ты мастер отношений множеств!";
  else if (percent >= 80) msg = "📖 Отлично! Ты хорошо понимаешь отношения объектов!";
  else if (percent >= 60) msg = "📖 Неплохо! Повтори свойства отношений — и будет 100%!";
  else msg = "📖 Не сдавайся! Запусти игру снова, чтобы прокачать тему!";
  document.getElementById("resultMotivation").innerHTML = msg;
  document.getElementById("startScreen").classList.add("hidden");
  document.getElementById("gameScreen").classList.add("hidden");
  document.getElementById("resultScreen").classList.remove("hidden");
  if (percent === 100) launchConfetti();
}

function launchConfetti() {
  const canvas = document.createElement("canvas");
  canvas.className = "confetti-canvas";
  canvas.width = window.innerWidth;

```

```

        canvas.height = window.innerHeight;
        document.body.appendChild(canvas);
        const ctx = canvas.getContext("2d");
        let particles = [];
        for (let i=0;i<120;i++) {
            particles.push({x: Math.random()*canvas.width, y: Math.random()*canvas.height-100,
            vx:(Math.random()-0.5)*6, vy:Math.random()*7+4, size:Math.random()*5+4, color:`hsl(${Math.random()*360},
            80%, 65%)`});
        }
        function animate() {
            if (!canvas.parentNode) return;
            ctx.clearRect(0,0,canvas.width,canvas.height);
            let alive = false;
            for(let p of particles){
                ctx.fillStyle = p.color;
                ctx.fillRect(p.x,p.y,p.size,p.size);
                p.x+=p.vx;
                p.y+=p.vy;
                if(p.y < canvas.height+80) alive=true;
                if(p.y>canvas.height+60) p.y=-60;
                if(p.x>canvas.width+60) p.x=-60;
                if(p.x<-60) p.x=canvas.width+60;
            }
            if(alive) requestAnimationFrame(animate);
            else canvas.remove();
        }
        animate();
        setTimeout(()=>{ if(canvas.parentNode) canvas.remove();}, 4000);
    }

    function resetGameToStart() {
        game = {
            answered: new Array(15).fill(false),
            correctMap: new Array(15).fill(false),
            consecutiveCorrect: 0,
            stars: 0,
            totalAnswered: 0,
            modalActive: false
        };
        renderPastryGrid();
        document.getElementById("startScreen").classList.add("hidden");
        document.getElementById("gameScreen").classList.remove("hidden");
        document.getElementById("resultScreen").classList.add("hidden");
    }

    document.getElementById("startBattleBtn").addEventListener("click", () => resetGameToStart());
    document.getElementById("playAgainBtn")?.addEventListener("click", () => resetGameToStart());
    renderPastryGrid();
    document.getElementById("gameScreen").classList.add("hidden");
    document.getElementById("resultScreen").classList.add("hidden");
    document.getElementById("startScreen").classList.remove("hidden");
</script>
</body>
</html>

```

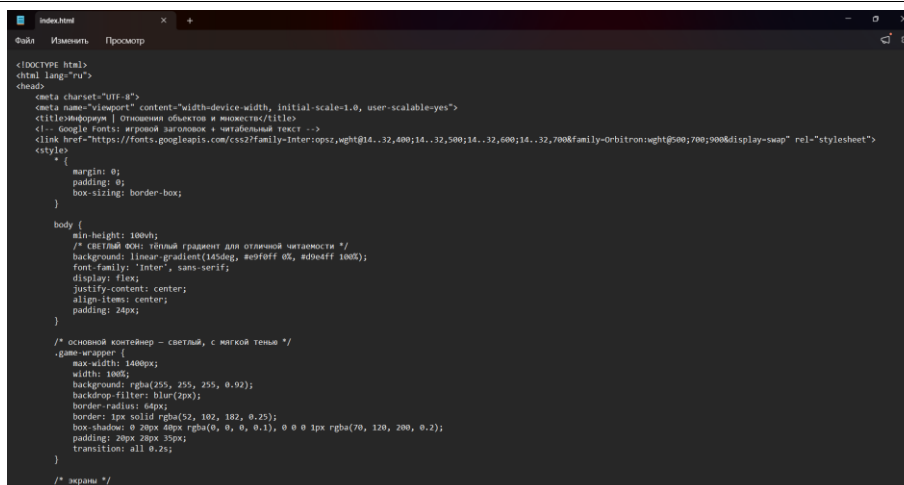


Рисунок 3. Создание файла

После сохранения файла открываем его в веб-браузере, на сайте отображается стартовый экран разработанной игры "ИНФОРИУМ" по теме "Отношения объектов и их множеств" с протоколом заданий, описывающим механику игры (15 вопросов-пирожных, система звезд за правильные ответы, блокировка отвеченных элементов), и интерактивной кнопкой запуска игрового сеанса.

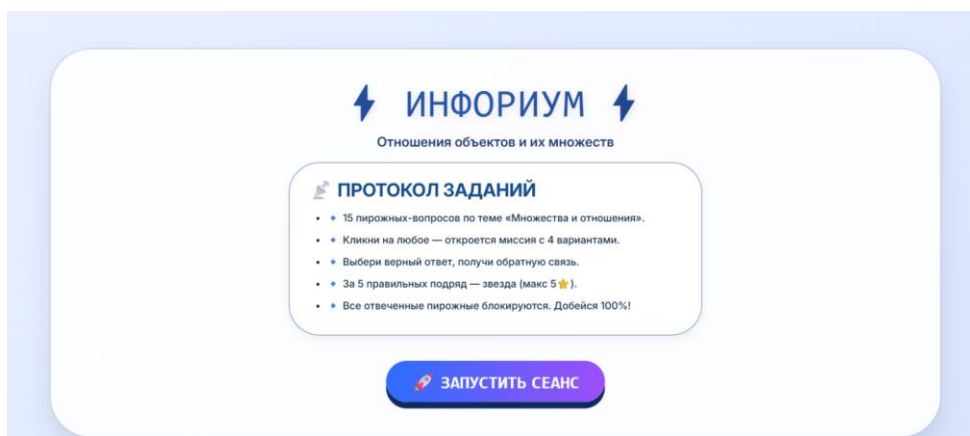


Рисунок 4. Главная страница

Затем, после запуска игрового сеанса отображается основное игровое поле, содержащее сетку из 15 интерактивных элементов в виде пирожных, каждое из которых представляет отдельное задание по теме «Множества и отношения», а также панель прогресса со счётчиком выполненных заданий (0/15) и индикатором достижений в виде пяти звёзд, активируемых при серии правильных ответов.

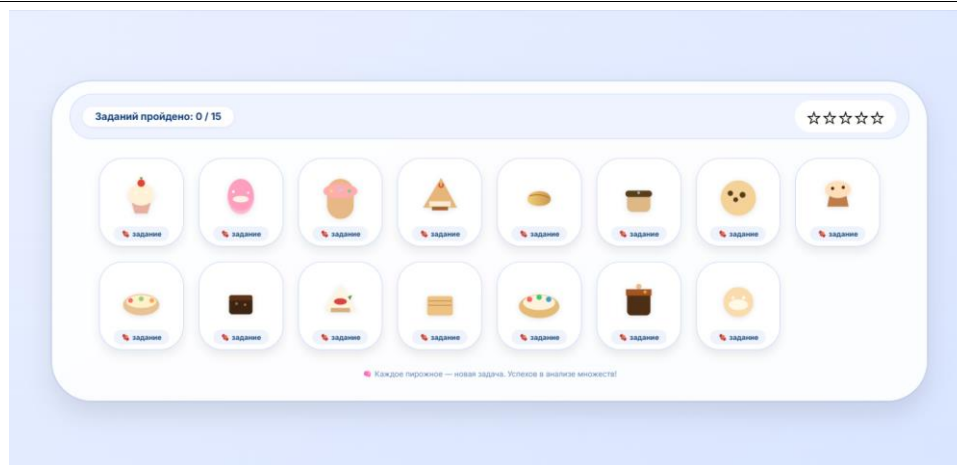


Рисунок 5. Задания

При клике на любое из пирожных открывается модальное окно с условием текущего задания (например, анализ свойств отношения «быть делителем» на заданном множестве), которое содержит четыре варианта ответа, кнопку для получения методической подсказки и элемент управления для закрытия окна.

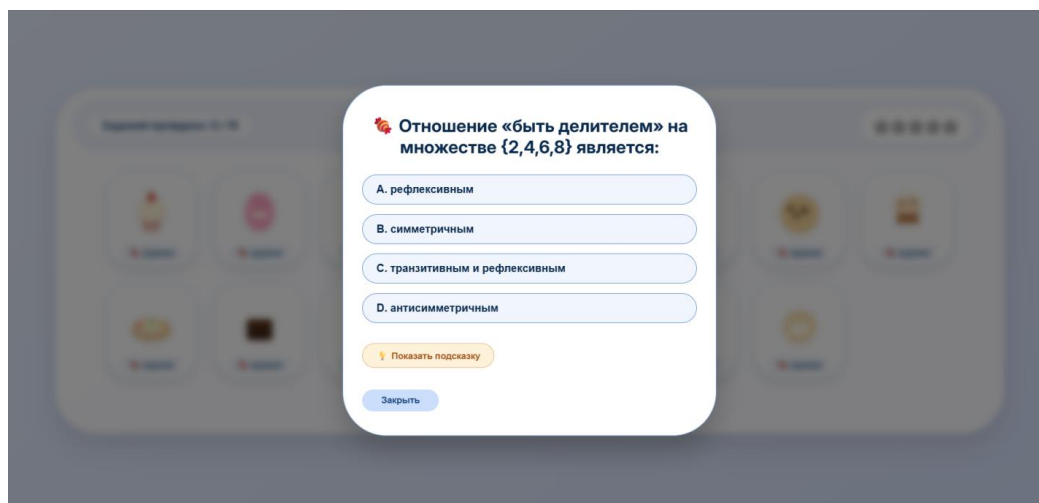


Рисунок 6. Выбор задания

После нажатия на подсказку пользователю отображается вспомогательная информация в виде текстового пояснения с определением или свойством математического понятия (например, «Число делит само себя, и если  $a|b$  и  $b|c$ , то  $a|c$ »), это реализует принцип педагогической поддержки и способствует формированию правильных представлений об изучаемых математических отношениях в процессе геймифицированного обучения.

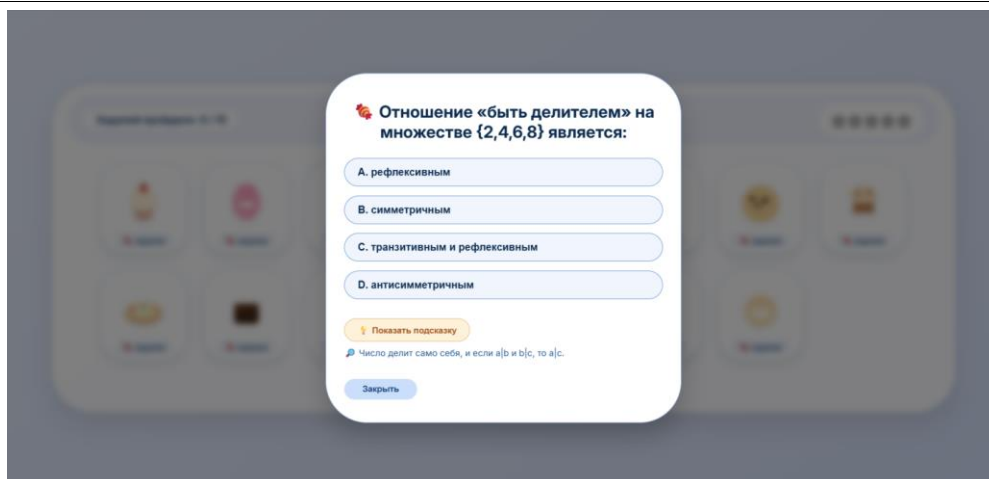


Рисунок 7. Подсказка

При правильном выборе ответа система отображает сообщение об успешном выполнении задания с визуальной индикацией (зелёная галочка и текст «Превосходно! Ответ верный.»), это обеспечивает немедленную положительную обратную связь, способствует закреплению правильных знаний и поддерживает мотивацию обучающегося к дальнейшему прохождению игровых заданий.

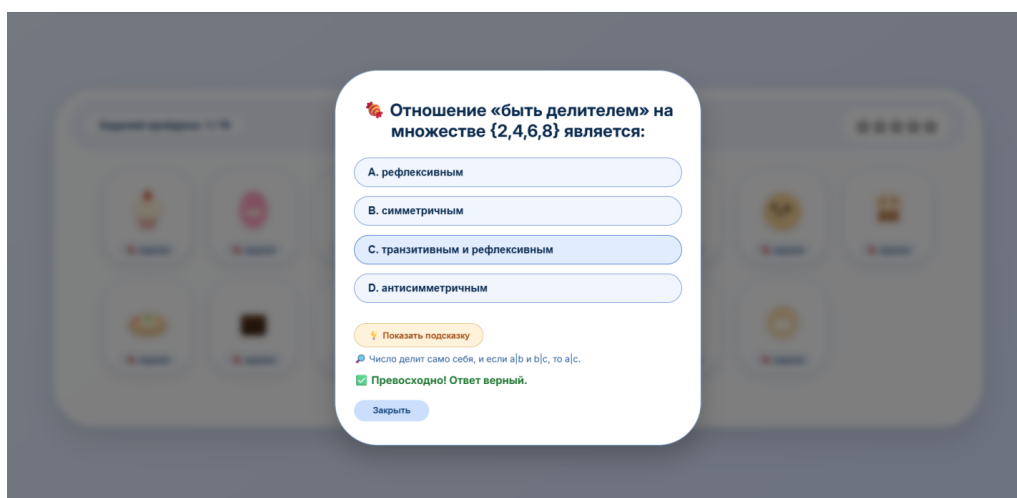


Рисунок 8. Правильный вариант ответа

После успешного выполнения задания соответствующий элемент на игровом поле визуально маркируется как решённый (зелёная подсветка, галочка и надпись «решено»), а счётчик прогресса в верхней части интерфейса автоматически обновляется, отражая текущее количество пройденных заданий (например, «Заданий пройдено: 1 / 15»), это обеспечивает наглядную динамику освоения материала и стимулирует завершение всех этапов игры.

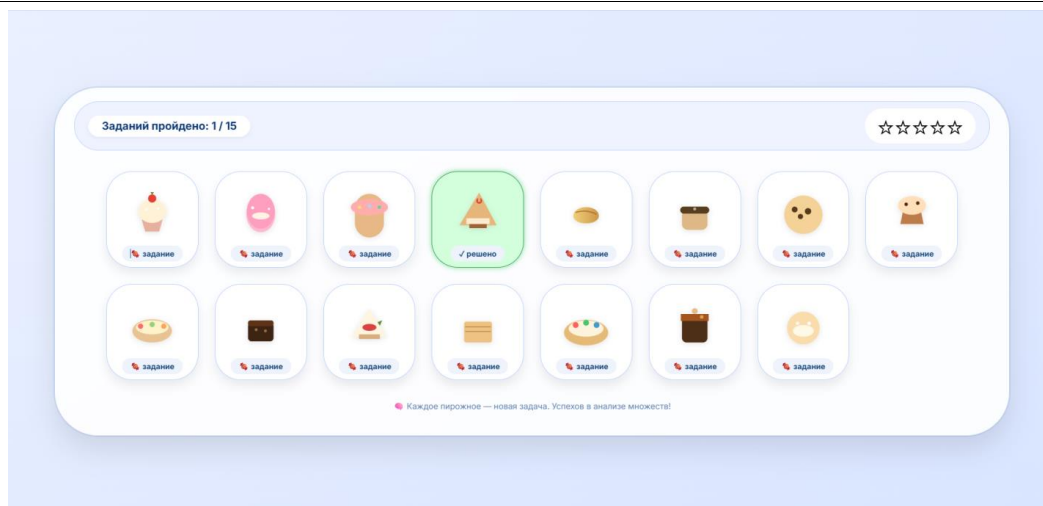


Рисунок 9. Маркировка элемента

Далее, при выборе неверного варианта ответа система генерирует уведомление об ошибке с демонстрацией эталонного ответа (в данном случае — «симметричное»), что реализует принцип немедленной коррекции знаний и способствует формированию устойчивых представлений о классификации бинарных отношений.

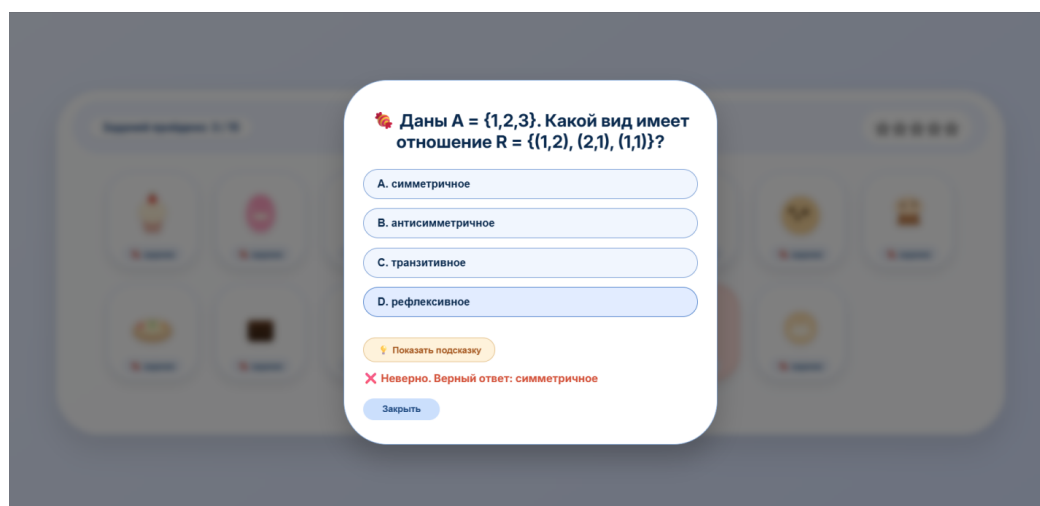


Рисунок 10. Ошибка

В случае ошибочного ответа соответствующий элемент игрового поля визуально маркируется красным цветом с пометкой «ошибка», а счетчик пройденных заданий остается неизменным, что реализует механизм обязательного повторного выполнения задачи и направлено на обеспечение полноты усвоения учебного материала.

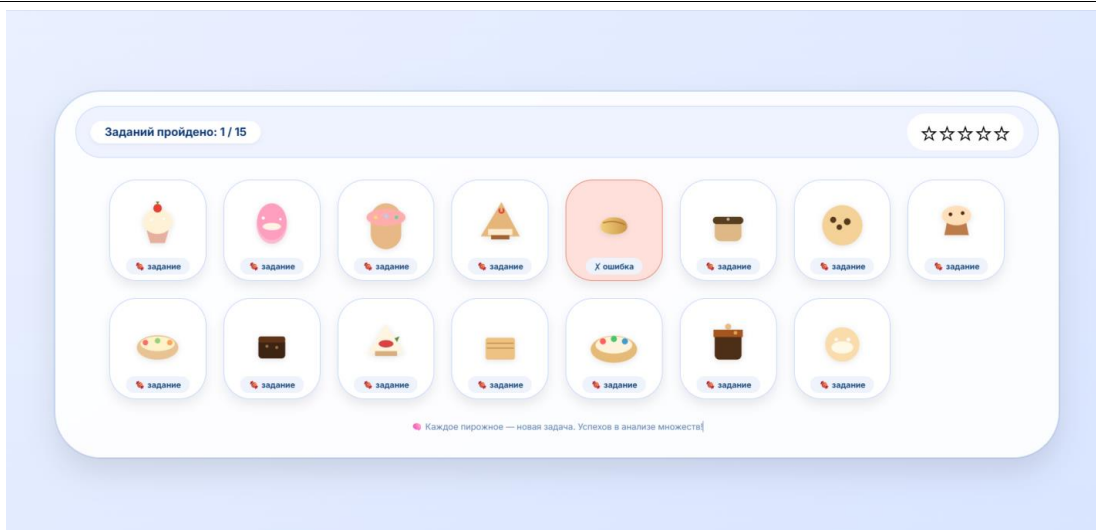


Рисунок 11. Ошибка

После того как пользователь завершает прохождение заданий, отображается итоговый экран отчета, демонстрирующий количественный результат в процентах и абсолютных значениях, содержащий мотивационное сообщение для повторного изучения темы и предоставляющий функциональную возможность перезапуска игрового сеанса для закрепления материала.

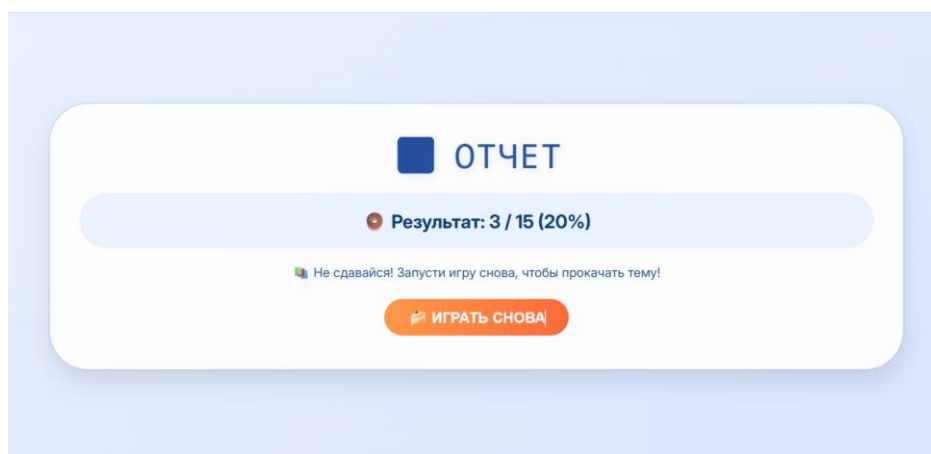


Рисунок 11. Результат

При достижении серии из пяти правильных ответов подряд система автоматически активирует первую звезду в панели достижений, что служит визуальным подтверждением высокого уровня усвоения материала и реализует механизм положительного подкрепления для поддержания учебной мотивации.

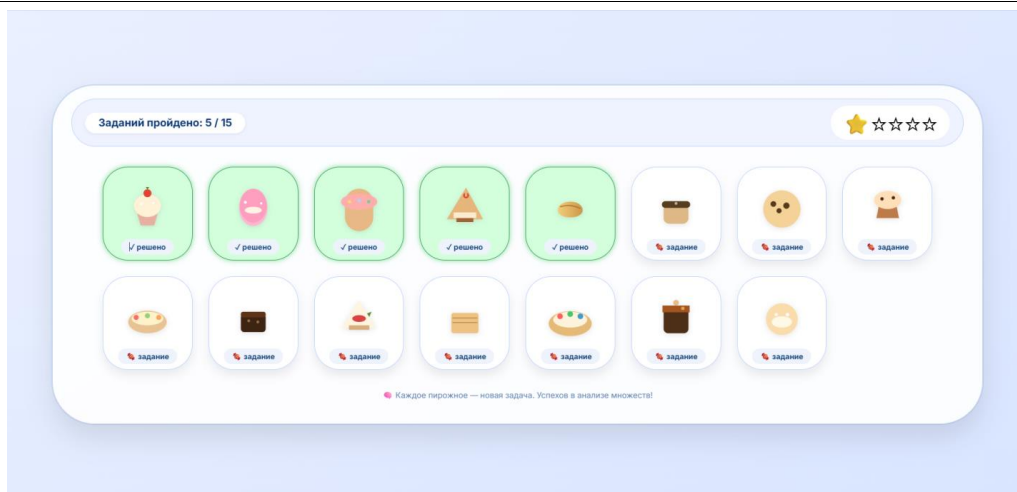


Рисунок 12. Первая звезда

В случае успешного выполнения всех заданий система отображает итоговый отчет со стопроцентным показателем результата (15/15), который сопровождается визуальными эффектами празднования и мотивационным сообщением, подтверждающим полное освоение темы, а также предоставляет пользователю функциональную возможность перезапуска игры для дальнейшего закрепления материала.

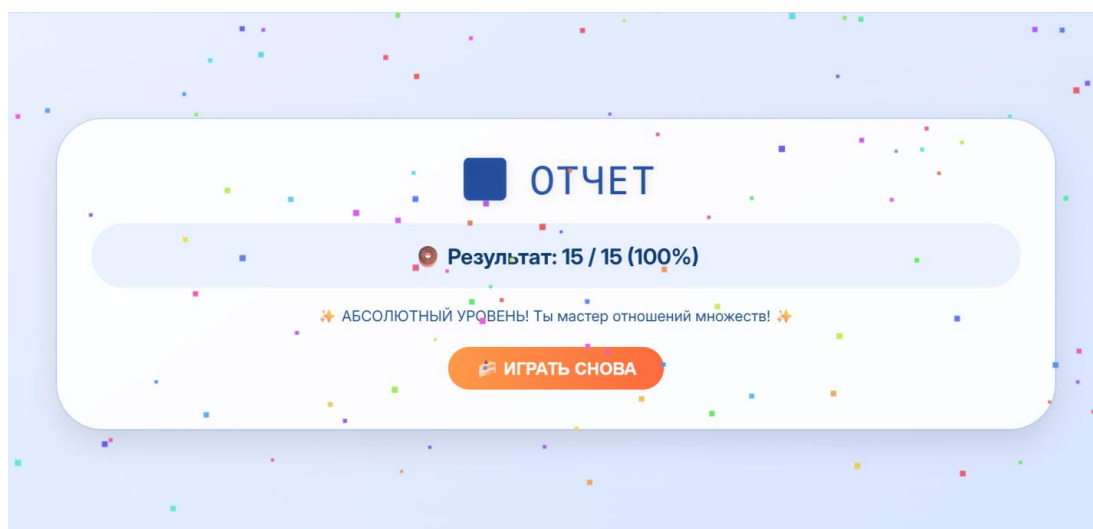


Рисунок 13. Успех

#### 4 Выводы

В данной работе детально рассмотрена методика создания интерактивной образовательной игры с использованием генеративного искусственного интеллекта DeepSeek. Методика охватывает полный цикл разработки, от регистрации в нейросети и составления структурированного промта до получения полностью функционирующего HTML/CSS/JavaScript-кода, адаптации игровой логики и публикации готового продукта.

Использование искусственного интеллекта DeepSeek делает процесс создания профессионального игрового образовательного контента доступным для пользователей без глубоких навыков программирования, в первую очередь

для преподавателей, методистов и организаторов учебных мероприятий. Генеративные возможности нейросети (создание семантической HTML-разметки, адаптивного CSS-дизайна, интерактивной JavaScript-логики, 15 уникальных SVG-изображений и системы геймификации) позволяют не только значительно сократить время на подготовку интерактивных материалов, но и обеспечивают вовлекающий игровой формат, напрямую влияющий на мотивацию и вовлеченность учащихся.

Практическая реализация методики на примере игры «Инфориум» по теме «Отношения объектов и их множеств» для 8–9 классов подтвердила её эффективность и тиражируемость для различных учебных дисциплин и возрастных категорий. Таким образом, генеративные нейросетевые модели становятся важным инструментом образовательной геймификации, способствующим развитию цифровых педагогических компетенций и отвечающим на вызовы современного образования, связанные с цифровизацией, необходимостью удержания внимания и адаптацией контента под интерактивные форматы. Освоение и внедрение подобных решений представляет собой закономерный шаг в эволюции методов обучения в условиях повсеместного использования цифровых технологий.

### Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С. 12-16.
2. Фёдоров И. А., Кузьменкова Е. Ю. Разработка игрового приложения «league of football heads» с использованием html5, css3, javascript //Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях. 2018. С. 311-311.
3. Королук Я. А., Куликова Н. Ю. Опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайнобучения школьников информатике //XXV Всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета. 2023. С. 48-55.
4. Размологов Д. М., Панина А. С., Захаров П. В. Использование html5 для разработки электронного учебника //Современные тенденции развития науки и технологий. 2016. №. 4-4. С. 92-94.
5. Шегаев И. С., Полозов А. Г. Новейший политологический глоссарий: основные категории, понятия, термины. Общество с ограниченной ответственностью Издательство Перо, 2015. 242 с.