

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

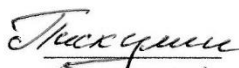
|                        |                                                                    |
|------------------------|--------------------------------------------------------------------|
| Факультет              | Кафедра                                                            |
| вычислительной техники | «МОиПЭВМ»                                                          |
| Направление подготовки | 09.04.02 – «Информационные системы и технологии»                   |
| Магистерская программа | Проектирование, разработка и эксплуатация<br>информационных систем |

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему

**Методика разработки программного обеспечения эволюционными  
методами**

Студент

  
(подпись, дата)

Пискулин Дмитрий Евгеньевич  
(ФИО полностью)

Руководитель

\_\_\_\_\_  
(подпись, дата)

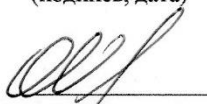
Кольчугина Елена Анатольевна  
(ФИО полностью)

Нормоконтролер

\_\_\_\_\_  
(подпись, дата)

Такташкин Денис Витальевич  
(ФИО полностью)

Рецензент

  
(подпись, дата)

Шеланков Олег Евгеньевич  
(ФИО полностью)

Работа допущена к защите (протокол заседания кафедры от \_\_\_\_\_ № \_\_\_\_\_)

Заведующий кафедрой

\_\_\_\_\_  
(подпись)

Макарьчев П.П.  
(фамилия, инициалы)

Работа защищена с отметкой \_\_\_\_\_ (протокол заседания ГЭК от \_\_\_\_\_ № \_\_\_\_\_)

Секретарь ГЭК

\_\_\_\_\_  
(подпись)

Попова Н.А.  
(фамилия, инициалы)

## Содержание

|       |                                                                 |    |
|-------|-----------------------------------------------------------------|----|
| 1     | Анализ предметной области исследования.....                     | 7  |
| 1.1   | Основные определения и применения эволюционных методов.....     | 7  |
| 1.2   | Эволюционные методы в разработке Web-интерфейсов .....          | 8  |
| 1.3   | Обоснование выбора объекта исследования.....                    | 10 |
| 1.4   | Постановка задачи.....                                          | 10 |
|       | Выводы .....                                                    | 12 |
| 2     | Информационная система для генерации дизайна Web-страницы ..... | 14 |
| 2.1   | Основная гипотеза предмета исследования.....                    | 14 |
| 2.2   | Взаимодействие пользователя с ИГА .....                         | 16 |
| 2.3   | Внедрение эволюционного алгоритма в язык CSS.....               | 18 |
| 2.4   | Обзор аналогов .....                                            | 22 |
| 2.5   | Описание информационной системы.....                            | 32 |
| 2.5.1 | План описания информационной системы .....                      | 32 |
| 2.5.2 | Взаимодействие компонентов системы .....                        | 33 |
| 2.5.3 | Концептуальная модель системы и функциональные требования ..... | 36 |
| 2.5.4 | Стадии разработки информационной системы .....                  | 45 |
|       | Выводы .....                                                    | 50 |
| 3     | Анализ работы эволюционных алгоритмов.....                      | 52 |
| 3.1   | Концептуальные модели работы генетических алгоритмов.....       | 52 |
|       | Выводы .....                                                    | 56 |
|       | Заключение .....                                                | 57 |
|       | Список использованных источников .....                          | 58 |
|       | Приложение А. Схемы эволюционных алгоритмов.....                | 61 |
|       | Приложение Б. Сгенерированные дизайн-макеты .....               | 65 |

## Введение

В середине XX века основные информационные исследования велись в области искусственного интеллекта. Изучение данной области основывалось на трёх методах моделирования: классическое моделирование, эвристическое программирование и эволюционное моделирование.

Классический подход предусматривал изучение непосредственно модели головного мозга человека. К таким моделям относятся нейронные сети. Первая компьютерная модель нейронных сетей была разработана Уоренном МакКаллаком и Уолтером Питтсом, которая базировалась на теории деятельности головного мозга человека и математических алгоритмов.

Методология эвристического программирования опирается на решения задач, которые основаны на опыте принятия решений. Главной особенностью задач такого типа является уменьшение вариантов, необходимых при использовании метода проб и ошибок. Однако при таком варианте поиска оптимальных решений можно пропустить более рациональный вариант.

В основу эволюционного моделирования положена идея заменить процесс моделирования человеческого интеллекта на процесс моделирования эволюции интеллекта. Это подразумевает то, что разумное поведение предусматривает сочетание способности предсказывать состояние внешней среды с умением подобрать реакцию на каждое предсказание, которое наиболее эффективно ведет к цели [1]. Другими словами, основной акцент сводится к адаптации искусственного интеллекта под определённые внешние условия.

Методология эволюционного моделирования получила широкое развитие благодаря доктору Лоуренсу Дж. Фогелю в начале 60-х годов прошлого века. В докладе конгрессу США Фогелем была представлена концепция эволюционного программирования. Концепция заключалась в моделировании непосредственно процесса эволюции, используя сам процесс в качестве транспортного средства для получения разумного поведения [2].

В дальнейшем на основе этой концепции эволюционного моделирования стали создаваться эволюционные алгоритмы. Идея эволюционных алгоритмов

заимствована у живой природы и состоит в организации эволюционного процесса, конечной целью которого является получение решения в сложной задаче оптимизации.

В 1975 году Джоном Холландом была предложена концепция генетических алгоритмов, которые основывались на принципах естественного отбора Ч. Дарвина [3]. Предложенные алгоритмы моделируют такие процессы, как: мутации, наследования определённых признаков от родителей, приспособление к окружающей среде и отбор наиболее сильных особей. Функциональность генетических алгоритмов основывается на поиске лучших решений с помощью наследования и усиления полезных свойств множества объектов определенного приложения в процессе имитации их эволюции. Постепенно эволюционные алгоритмы стали применяться во многих оптимизационных задачах.

С развитием информационных технологий и созданием сети интернет начали появляться первые примитивные сайты, которые открыли для пользователей доступ к новейшей информации. Со временем внешний вид сайтов претерпевал значительные изменения в плане внешнего вида и функциональных возможностей. Web-сайты стали большим подспорьем для реализации задач оптимизации, а именно для поиска новых дизайнерских решений.

Идея использования генетических алгоритмов для дизайна Web-страниц мотивирована системой, созданной Карлом Симсом, которая использует генетические алгоритмы для создания 2D текстур [4]. Мотивация его работы была в основном художественной, в результате чего художник непосредственно определял пригодность каждого решения путем визуального осмотра. Его система позволяла случайным образом исследовать пространство текстур с решениями, сходящимися на основе симпатий и антипатий художника.

Эволюционный механизм может улучшить процесс проектирования, позволяя дизайнерам и проектировщикам одновременно сравнивать и оценивать решения динамическим и интуитивно понятным способом.

Хотя цели Web-дизайна и давно созданного вычислительного эволюционного процесса поиска различны, основная проблема, то есть размещение и структура элементов, одна и та же.

Проблема с визуальной составляющей Web-сайтов имеет многомерное пространство проблем. Генетические алгоритмы могут существенно упростить решение задач, которые связаны с оптимизацией интерфейса сайта и помогут сократить время на его проектирование. Поэтому вопрос о необходимости внедрения генетических алгоритмов в Web-разработку является на данный момент актуальной.

**Цель диссертации** состоит в изучении и дальнейшем применении эволюционных методов для разработки Web-сайтов. На основе изученных методов выработать механизм работы генетических алгоритмов для проектировки дизайна сайта. Показать эффективность, а также быстроту применяемых эволюционных методов, для ускорения работы по нахождению лучших дизайнерских вариантов для Web-страниц.

**Задачи исследования.** Основная последовательность выполняемых задач:

- изучение эволюционных алгоритмов, связанных с Web-разработкой;
- разработка концептуальной модели информационной системы с использованием предложенных генетических алгоритмов;
- разработка генетических алгоритмов для формирования новых дизайнерских решений.

**Объектом исследования** была выбрана область в Web-разработке, отвечающая за внешний вид сайта.

**Предметом исследования** являются эволюционные методы и алгоритмы, использованные в разработке сайтов для создания новых вариантов дизайнерских решений.

**Методы исследования.** В соответствии с поставленными целями исследования, работа включает в себя современные методы изучения, связанные с эволюционными поиском и решением задач оптимизации.

**Научная новизна** работы заключается в следующем:

- эволюционные методы позволят автоматизировать работу по

созданию удобного внешнего вида Web-ресурсов;

- разработанные генетические алгоритмы позволят получать большое количество новых вариантов решений для нахождения лучшего дизайна;
- работа эволюционных алгоритмов в несколько раз сокращает время работы над поиском проектных решений;
- применение эволюционных методов увеличивают производительность новых Web-страниц.

**Практическая значимость исследований.** На основе концептуальной модели информационной системы в дальнейшем можно создать программное обеспечение, которое позволит пользователям в сферах Web-разработки за короткое время создавать новые макеты для будущих Web-сайтов.

**Основные положения, выносимые на защиту:**

- эволюционные методы, которые подходят для решения задач в области оптимизационных задач, связанные с созданием сайтов;
- модель информационной системы, показывающая процесс работы интерфейса по генерации новых макетов;
- созданные генетические алгоритмы для получения новых вариантов дизайна сайта;
- анализ эффективности работы системы на основе эволюционных алгоритмов.

**Публикации.** По теме диссертационного исследования опубликована одна научная статья.

**Структура и объем работы.** Работа содержит: введение, три тематические главы, заключение, список литературы, два приложения, одну таблицу, 29 рисунков основной части и 8 рисунков из приложений. Объем основного текста – 60 страниц. Список литературы представлен на двух страницах и содержит 21 наименование.

## **1 Анализ предметной области исследования**

### **1.1 Основные определения и применения эволюционных методов**

Современное развитие технологий предполагает новые механизмы проектирования информационных систем. На смену программам, которые имеют характер ручного управления, появляются системы, функционирующие на основе интеллектуальных технологий программирования. Основным эволюционным методом разработки самоорганизующихся информационных систем служит генетическое программирование, которое предложил ученый Дж Фогель. Оно было усовершенствовано учёным Д. Коза [5] на основе модели классического эволюционного алгоритма.

Генетические алгоритмы — это адаптивные методы поиска, которые в последнее время используются для решения задач оптимизации. В них используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде и основные понятия линейной алгебры [6].

Генетическое программирование хорошо зарекомендовало себя с точки зрения решения практически значимых задач [7]. В зависимости от условия задачи результат значения целевой функции у оптимизационных задач может быть разным. В основном эти задачи сводятся к поиску новых вариантов решений оптимизации. Поиск новых решений осуществляется путём технологии локального поиска [8]. Различают следующие виды поиска новых вариантов решений [8]:

- одномерный поиск, в котором различают метод пассивного и последовательного поиска;
- численные методы;
- статические методы оптимизации.

Каждый из предложенных методов имеют индивидуальный характер поиска решений для задач оптимизации. В рамках данной научной диссертации рассматривается метод одномерного поиска решения задач с помощью

древовидной структуры, так как он наилучшим образом применяется в задачах оптимизации, связанных с разработкой Web-интерфейсов.

Действия программ, основанные на методе генетического программирования, под действием эволюционных алгоритмов в большинстве случаев, могут порождать решения, которые далеки от необходимого результата. Однако, факт получения пользователем некорректных решений не препятствует дальнейшему изучению возможности внедрения эволюционных методов в разработку Web-интерфейса информационной системы.

## **1.2 Эволюционные методы в разработке Web-интерфейсов**

Информационные системы, спроектированные на основе методов эволюции, позволяют автоматизировать выполнение конкретных задач. Большинство задач информационных систем носит оптимизационный характер. Результат таких задач должен выполнять необходимые критерии и удовлетворять потребности пользователя.

Решение задач по поиску новых оптимизационных результатов происходит через выполнения специальных алгоритмов.

Генетические алгоритмы представляют собой определённые шаги, по которым выполняется процесс формирования новых решений. Основанные на эволюционной модели процесса естественного отбора генетические алгоритмы находят решения задачи посредством получения новых, наиболее приспособленных результатов (индивидуумов).

Из многочисленных вариантов, где возможно применение эволюционных алгоритмов, можно выделить сферу, связанную с Web-разработкой. Дизайн сайта представляет собой различные способы расположения элементов на странице. Это делает каждый сайт по-своему уникальным.

В статье [9] авторами Е. А. Кольчугиной и К. В. Заваровским рассматривается идея получения новых вариантов Web-интерфейса в языке HTML.

Был предложен способ, к котором HTML-код представлялся в виде древовидной хромосомы. Такой вид представления кода позволяет генетическим

операторам порождать всевозможные решения, связанные с оптимизацией Web-интерфейса.

Данный метод разработки пользовательских интерфейсов позволяет частично автоматизировать данный процесс [9], а также помочь пользователям, которые специализируются на дизайне применять новые решения в своих работах.

На основе исследований авторов строится основная гипотеза данной научной работы. Цель диссертации расширить исследования авторов путём применения эволюционных методов в формировании дизайна Web-сайтов: сформировать схемы эволюционных алгоритмов и представить описание концептуальной модели информационной системы, которая находит новые варианты дизайн макетов.

Исходя из рассматриваемой статьи [9] можно сделать вывод, что тема исследований эволюционных методов разработки Web-интерфейсов является актуальной с точки зрения решения задачи оптимизации.

Тема применения генетического программирования в Web-интерфейсах из [9] является актуальной в вопросе рассмотрения проблем разработки пользовательских интерфейсов в автоматизированных системах [10, 11]. Внедрение эволюционных методов в Web-интерфейсы рассматриваются как способ прототипирования Web-приложений.

Данный факт использования генетического программирования для создания прототипа будущего приложения имеет место быть, так как на основе генетических операций, проводимых в эволюционных алгоритмах можно получить новые варианты макетов, по которым будет разрабатываться информационная система. В рассматриваемом случае прототипирование в Web-приложениях посредством эволюционных методов является мало исследованной областью.

В рамках научной исследовательской совмещаются вопросы о получении новых прототипных макетов Web-дизайна и создании концептуальной модели информационной системы, которая генерирует эти самые макеты.

### **1.3 Обоснование выбора объекта исследования**

На сегодняшний день профессия дизайнера вышла на первый план. Сейчас ни одно средство массовой информации не обходится без качественной рекламной интеграции будь то баннер или интерактивный интернет-магазин. Первое, что всегда бросается в глаза потенциальному пользователю – это оформление: шрифт, информационные секции, навигация меню и т.д. Нужно понимать, что на составление хорошего дизайнерского оформления требуется очень много сил и времени. Также нужно заметить, что дизайн неотрывно связан от Web-разработки и идёт с ним плечом к плечу. Основной акцент в научной диссертации сделан на совмещении исследований в сфере эволюционных методов с разработкой Web-сайта.

Во-первых, использование эволюционных методов проектирования в разработке сайта предоставляет разработчику широкий выбор вариантов макетов, которые были созданы с помощью операторов генетического алгоритма. В зависимости от поставленной задачи разработчик выберет тот вариант, который наиболее точно отражает суть выполнения технического задания.

Во-вторых, генерирование новых макетов значительно сэкономит время на разработку страниц для будущего Web-сайта. Это позволит разработчику оставшееся время доработать другие составляющие Web-страницы.

### **1.4 Постановка задачи**

Целью магистерской диссертации является создание концепта информационной системы с помощью диаграмм процессов. Задачей исследования является внедрение в язык каскадных таблиц стилей CSS генетических алгоритмов для поиска оптимального решения при разработке макетов дизайна Web-сайтов.

Основные этапы научного исследования:

- сформировать схему архитектуры информационной системы;
- сформировать блок-схемы эволюционных алгоритмов;

- разработать концептуальную модель информационной системы, которая будет генерировать новые варианты исходного макета будущего Web-сайта;
- показать последовательность выполнения действий генетических алгоритмов на основе диаграмм процессов;
- показать результат работы эволюционных алгоритмов.

Система должна быть представлена для пользователей, занимающихся проектированием дизайна макетов для будущего Web-сайта. Главными критериями программы будут являться:

- последовательное и детерминированное выполнение всех действий, предусмотренных программой;
- быстрое выполнение поставленных задач.

Система на основе ГА должна повысить эффективность работы над проектированием визуального вида будущего сайта, с помощью поиска и оптимизации. Далее следуют основные требования, предъявляемые к информационной системе.

1. Правильность работы. Система с помощью интерактивного генетического алгоритма находит новые решения, связанные с внешним видом будущего сайта.
2. Понятность функционирования. Функции, которые выполняет программа полной мере должны выполнять определённые действия.
3. Измеримость. Информационная система должна эффективно экономить время пользователя.
4. Непротиворечивость. Все вышеуказанные требования, должны быть взаимосвязанными.

Внедрение ГА в язык CSS должно в значительной мере ускорить время проектирования макетов, снизить временные затраты на выбор оптимального решения, позволит значительно уменьшить затраты рабочего времени. В результате работы генетического алгоритма пользователь будет иметь возможность оценить сгенерированный макет и сохранить результат действия алгоритмов.

Итоговым результатом научной диссертации будет является дизайн-макеты, которые получились под действием работы эволюционных алгоритмов.

## **Выводы**

В первой главе научной диссертации, были сформулированы некоторые основные термины эволюционных процессов: дано определение генетическому алгоритму. Также были представлены основные методы поиска. Далее была описана тематика предметной области и предполагаемый объект его рассмотрения. Кратко была сформулирована проблема, связанная с разработкой интерфейсов в Web-дизайне, и были намечены пути возможного решения.

Цель обзорной части состояла в том, чтобы показать необходимость и большую значимость внедрения эволюционного программирования и генетических алгоритмов для поиска оптимальных решений, связанных с дизайном страниц. Сформулирована основная цель научной диссертации и указаны последовательные действия для её решения.

В предложенной концептуальной модели информационной системы будет описан процесс функционирования генетических алгоритмов для создания Web-страниц. Также потенциальный пользователь (дизайнер или верстальщик) имеет возможность оценить результат действия алгоритмов и сохранить промежуточный результат для дальнейшей работы.

В исследовании предлагается использовать каскадные таблицы стилей CSS, чтобы помочь процессу оптимизации дизайна Web-страницы. Этот метод поможет дизайнерам редактировать и улучшать стиль веб-сайта, чтобы соответствовать его эстетическому и функциональному представлению.

Следующим этапом в исследовании научной работы станет детальное описание взаимодействия генетических алгоритмов в языках гипертекстовой разметки.

Последовательность решения задач во второй главе включает в себя следующие шаги:

- формулировка основной гипотезы исследования для разработки программного обеспечения;

- выстраивание рабочей концепции на основе предложенной методики, которая является отправной точкой в построении генетических алгоритмов для разработки программного обеспечения;
- описание функциональных возможностей информационной системы;
- выделение теоретического и практического аспекта значимости процесса работы информационной системы.

## **2 Информационная система для генерации дизайна Web-страницы**

### **2.1 Основная гипотеза предмета исследования**

Идея для написания магистерской диссертации возникла из моего интереса к поиску сходства между искусством и наукой, которая изучает эволюционные методы исследования.

В науке, теория эволюции утверждает, что те, кто наиболее приспособлен к данной природной среде, могут выжить и эволюционировать до следующего поколения [12].

Web-сайт представляет собой виртуальное пространство, где различного рода функции и навигации взаимодействуют между собой так же, как в реальной окружающей среде особи взаимодействуют между собой. Например, при определённом нажатии на кнопку происходит некое видоизменение интерфейса. Таким образом кнопка через действие пользователя взаимодействует с какой-то частью интерфейса на сайте.

Рассматривая дизайн, как «проблему оптимизации», нужно надеяться на то, что предложенный алгоритм может процедурно улучшить «популяцию» макетов, для того чтобы впоследствии выбрать из многообразия «потомков» тот макет страницы, который будет основой для Web-сайта.

Основная гипотеза заключается в следующем. Генетический алгоритм служит высокоэффективным методом поиска оптимальных решений сложных многомерных проблем с использованием концепции биологической эволюции. Языки гипертекстовой разметки, предназначенные для разработки Web-сайта, можно рассматривать как хромосомы, которые имеют наследственную информацию. Это информация заключена в тегах сайта, из которых строится сама структура Web-страницы.

Язык Html формирует каркас сайта и показывает, как будут располагаться элементы на странице. В статье о «Применении методов генетического программирования к разработке веб-интерфейсов» [9] и в статье [13] говорится о древовидной структуре Html. Операторы генетического алгоритма

применяются к узлам этого дерева, под действием которых получаются новые Web-страницы. Однако такой подход изменяет лишь саму структуру макета, не влияя на визуальное представление сайта.

Рассмотрим также другой язык гипертекстовой разметки. Если Html – это, так называемый, «скелет» сайта, то язык каскадных таблиц стилей (CSS) является его «мышцами», внешней оболочкой. Язык CSS связывается с HTML благодаря специальным классам. Для каждого тэга HTML класс является уникальным.

Существенное отличие между этими языками является то, что если применять алгоритм к HTML, то относительно работы операторов скрещивания и мутации будут меняться блоки секций сайта: навигация, галерея и т.д. В CSS под действием алгоритма, будут изменяться и структура секций сайта, и дизайн этих секций (блоков): цвет, размер шрифта, рамки и т. д. CSS придаст странице определённый вид, что сделает макет целостным.

Сами по себе каскадные таблицы стилей имеют блочную структуру, что позволяет сравнить их с линейными хромосомами. На рисунке 1 изображено сходство между каскадными таблицами стилей и схемой линейной хромосомы.

#### Блочная модель CSS

```
.class {
  атрибут 1: значение 1;
  атрибут 1: значение 2;
  атрибут 3: значение 3;
  ...
  атрибут n: значение n;
}
```

#### Схема линейной хромосомы

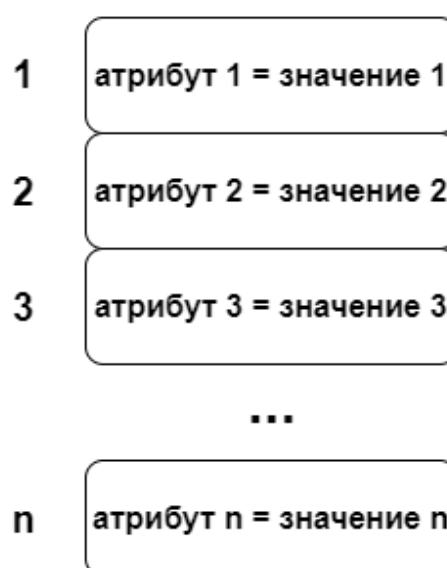


Рисунок 1 – Сравнение структуры CSS и линейной хромосомы

Исходя из сравнения блочной модели CSS и схемы линейной хромосомы можно сделать вывод о том, что каскадные таблицы стилей хорошо подходят для работы с эволюционными алгоритмами. Под действием эволюционных операторов стили CSS могут изменять форму макета Web-страницы и получать новый дизайн.

Поэтому гипотеза применение генетического алгоритма к языку CSS, поможет решить две задачи: получить хороший дизайн и изменить структуру сайта, тем самым отбрасывая более слабые варианты сгенерированных макетов. Работа эволюционных алгоритмов позволит пользователю оптимизировать процесс Web-разработки и оценить полученный результат.

## **2.2 Взаимодействие пользователя с ИГА**

В этом подразделе выстраивается фундаментальная концепция взаимодействия интерактивного генетического алгоритма с пользователем. Процесс взаимодействия позволит представить основную концепцию того, как ведёт себя генетический алгоритм в ответ на действия пользователя. Также выделяются этапы его взаимодействия с пользователем в работе программного обеспечения.

Интерактивный подход в первую очередь применяется к задачам проектирования [14]. Методика разработки информационной системы с применением эволюционных методов предусматривает тот факт, что пользователь имеет возможность вмешиваться в процесс работы генетического алгоритма и взаимодействовать с ним.

Взаимодействие пользователя опирается на изменение отдельно взятых элементов графического дизайна: фоновый цвет, начертания текста, тень и радиус рамки и т.д. Разработка успешного механизма взаимодействия пользователя с интерактивным генетическим алгоритмом обеспечит наилучший результат выполнения поиска оптимизационных решений.

Для наглядного представления как пользователь интерактивно взаимодействует с генетическим алгоритмом можно представить на рисунке 2.

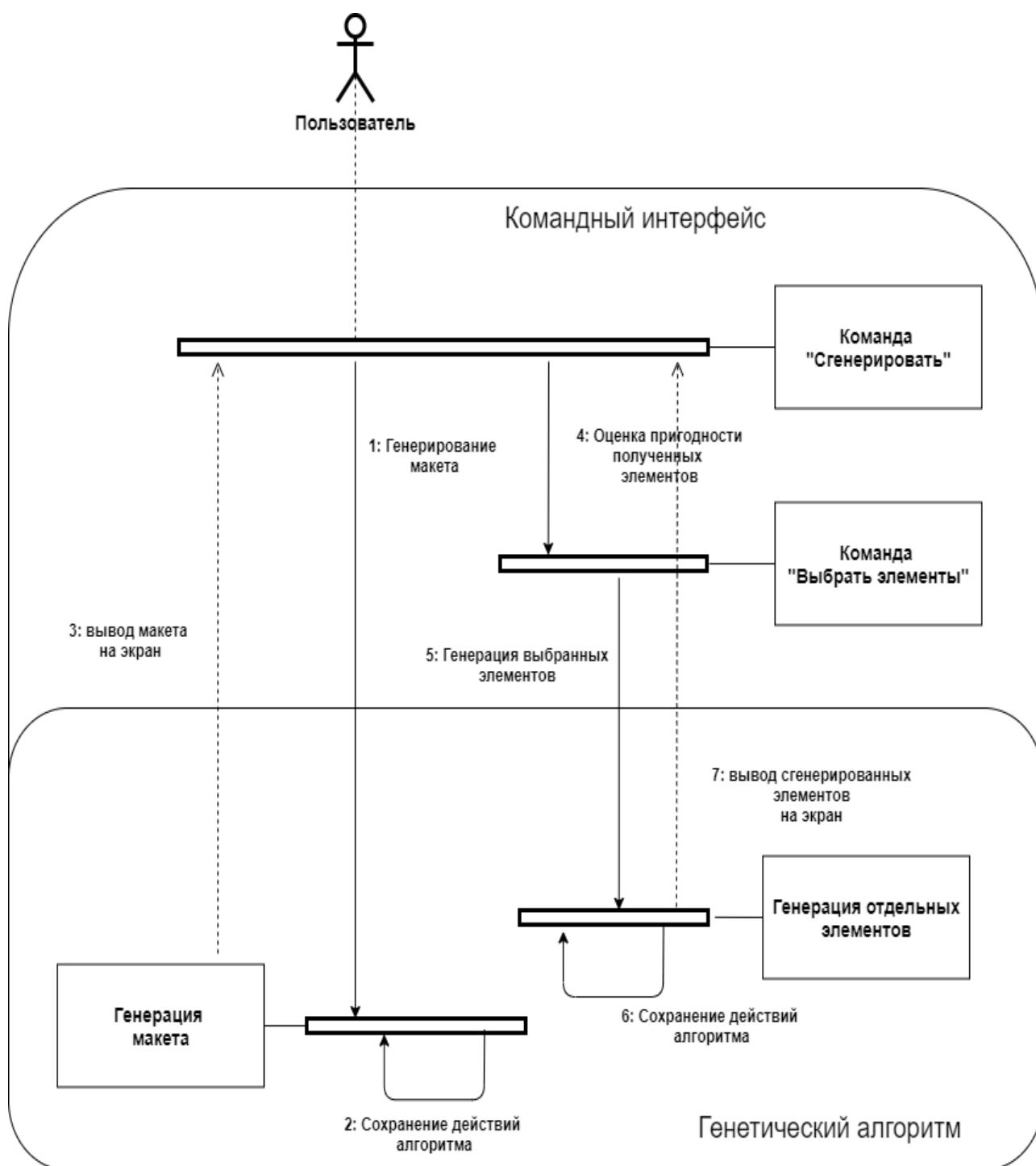


Рисунок 2 – Диаграмма взаимодействия пользователя с ИГА в программном обеспечении

Методика взаимодействия с ИГА в данной научной работе предполагает следующие подходы.

1. Пользователь загружает первоначальный макет и генерирует его.
2. Алгоритм под действием генетических операторов производит новый сгенерированный макет.

3. Новый макет выводится на экран, тем временем действия алгоритма сохраняются, чтобы при повторном использовании системы лучше оптимизировать работу.

4. Пользователь на основе своих эстетических предпочтений даёт оценку пригодности полученному макету.

5. При «неблагоприятной» генерации отдельно взятых индивидуумов (цвета текста, размера блоков и т.д.) пользователь через интерфейс сообщает алгоритму о том, что нужно сгенерировать отдельные участки макета.

6. Исходя из этого генетический алгоритм принимает команду от пользователя и акцентирует своё внимание на отдельных частях макета. При этом остальные части страницы остаются без изменения. Действия по генерации отдельных элементов сохраняются в памяти алгоритма.

Таким образом между пользователем и генетическим алгоритмом происходит «отзывчивое» взаимопонимание. При следующем использовании программного обеспечения генетический алгоритм будет учитывать пожелания пользователя и больше уделять внимания «проблемным» участкам макета.

Перейдём к рассмотрению языка гипертекстовой разметки с которым будет работать генетический алгоритм.

### **2.3 Внедрение эволюционного алгоритма в язык CSS**

Таблица стилей представляет собой написанный на языке CSS набор правил, применяемых к HTML-тегам документа, к которому эта таблица подключена. Правила эти задают визуальное представление содержимого HTML-тегов. По-другому говоря, как эти правила будут выглядеть на экране монитора (цвет, рамки, размер, отступы и т.д.). Благодаря языку CSS эти параметры не надо задавать отдельно для каждого тэга через его атрибуты (рисунок 3). Каскадные таблицы стилей позволяют отделять содержимое HTML-документа от описания в нём внешнего вида Web-страницы. Таким образом, благодаря этому разграничению достигается значительное удобство, компактность описания визуальных свойств HTML-элементов и оперативность их изменения.

## Дерево HTML

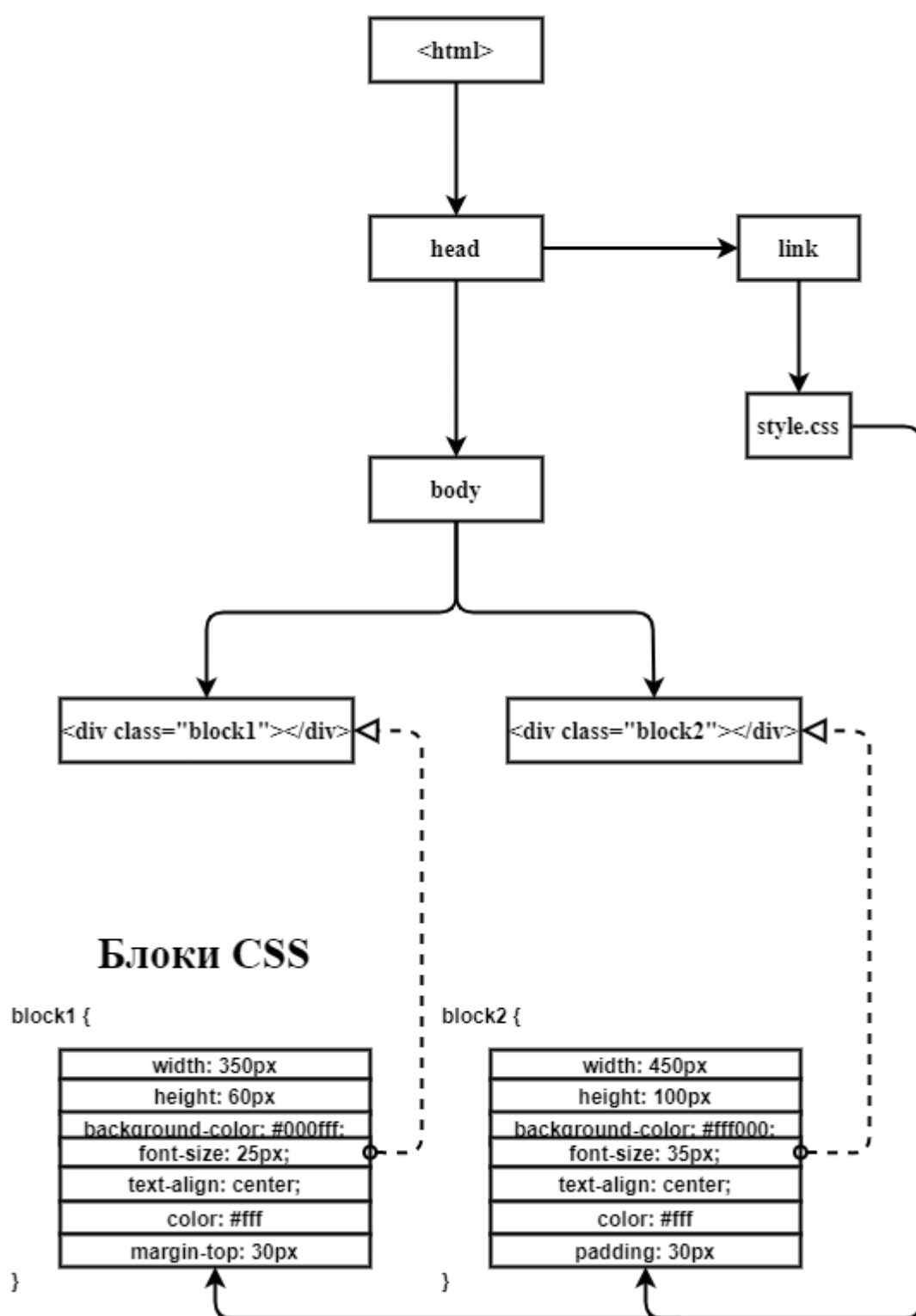


Рисунок 3 – Взаимодействие CSS-стилей с тэгами HTML

Стандарт CSS стал неотъемлемым дополнением языка гипертекстовой разметки HTML. Без применяемых стилей невозможно создать полноценный Web-интерфейс, потому что визуальная часть является основной в

формировании представления о хорошем сайте. Возможности каскадных таблиц стилей, применяемых к различным тегам языка HTML, имеют колоссальный потенциал. Он варьируется от выбора цветовой гаммы до различных сложных эффектов над элементами. Для нахождения новых комбинаций, направленных на получение новых дизайнерских решений, целесообразно использовать генетический алгоритм [15].

Работа генетического алгоритма в CSS представлена на рисунке 4. В качестве примера на рисунке изображена операция кроссинговера, которая производит обмен значений атрибутов.

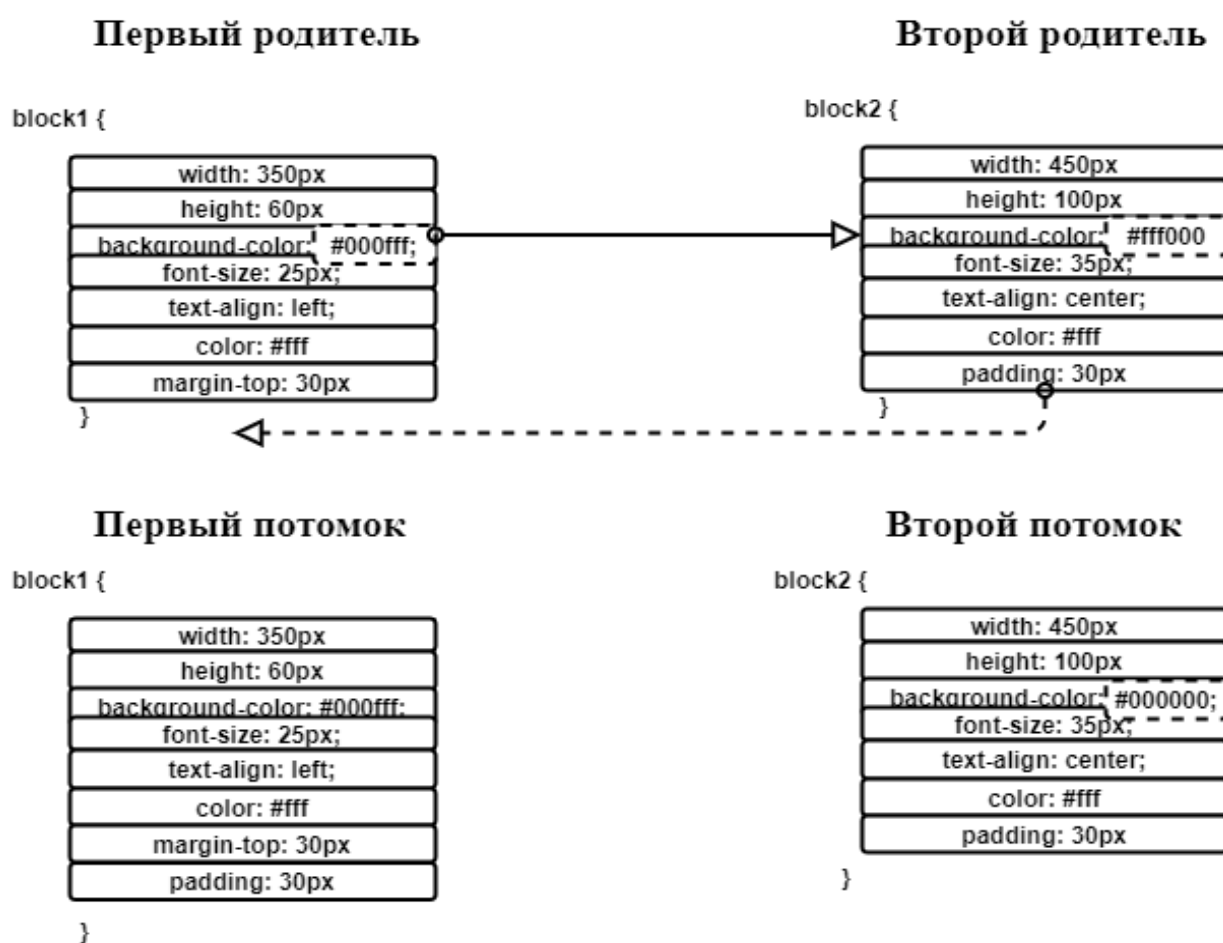


Рисунок 4 – Пример операции кроссинговера в языке CSS

На рисунке 5 изображена схема рабочего процесса, которая показывает, как генетические операции преобразуют таблицы стилей CSS в макете страницы.

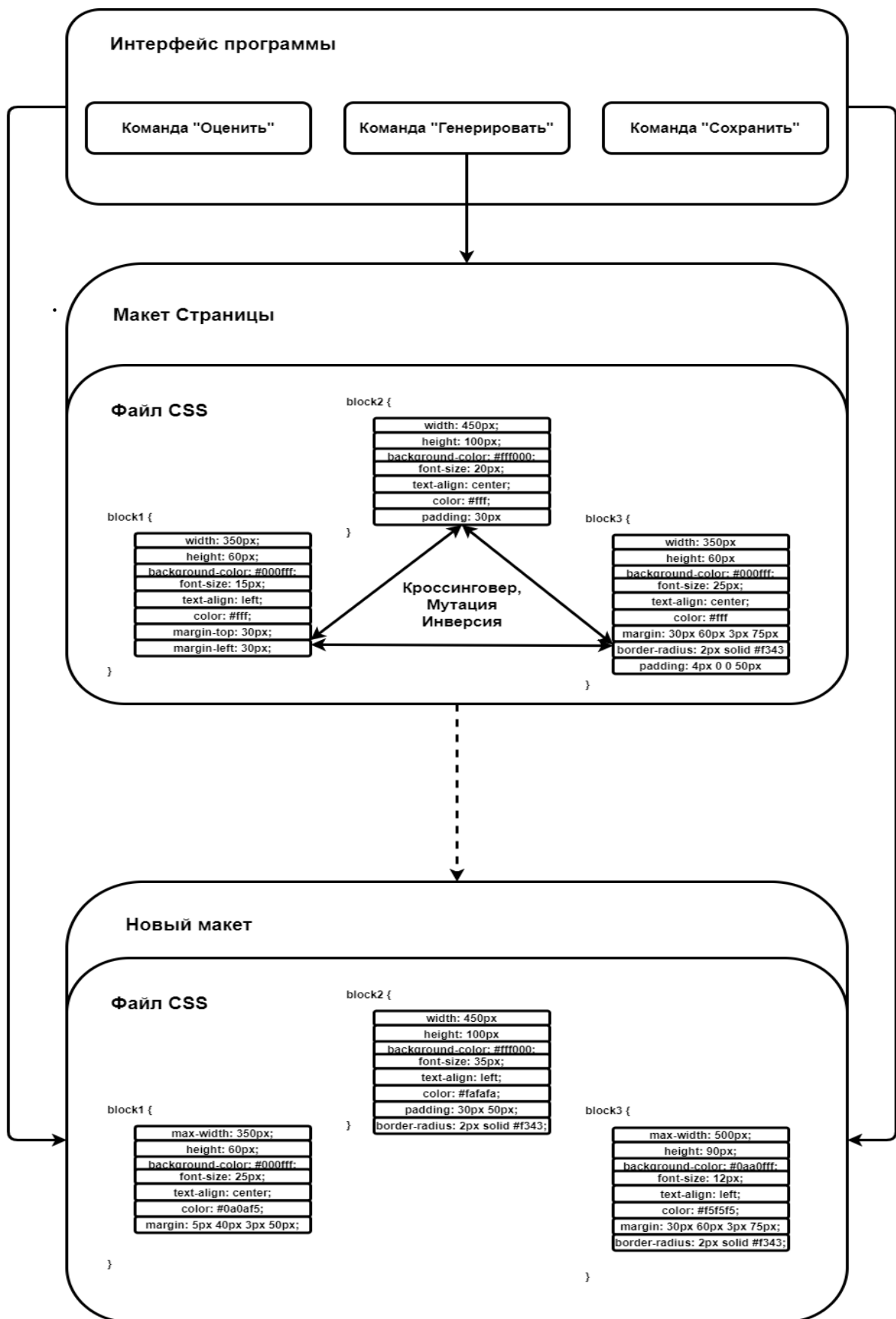


Рисунок 5 – Схема генерации CSS блоков в макете

Преимуществом включения CSS во взаимодействие с генетическим алгоритмом являются:

- 1) многообразие вариантов использования значений атрибутов: значения использованы в качестве хромосом, которые можно скрещивать и подвергать мутации;
- 2) значения атрибутов привязаны к тэгам HTML, которые связаны с определенным классом: это позволяет применять сгенерированные решения ко всему HTML-документу, где встречается изменяемый тэг;
- 3) каскадный язык стилей является более гибким и более приспособлен для взаимодействия с генетическим алгоритмом [15].

## 2.4 Обзор аналогов

Прежде чем приступить к описанию концептуальной модели информационной системы, которое генерирует новые макеты Web-страниц, нужно ознакомиться с другими аналогичными программными обеспечениями. Рассматриваемые аналоги работают через браузер, что позволяет в режиме реального времени провести соответствующий анализ приложения, выявить Основные плюсы системы и минусы.

1. Enjoy CSS – это онлайн CSS-генератор, который позволяет создавать разнообразные CSS-стили за короткое время при минимальных усилиях. Это приложение имеет удобный интерфейс, который состоит: из главного экрана, верхнего меню и бокового меню.

На главном экране приложения отображаются отдельные элементы для сайта. Этих элементов четыре: пустой блок, элемент ввода информации, блок с наполнением информацией, кнопка для навигации по сайту. Перечисленные элементы пользователь может создать с помощью бокового меню. В свою очередь боковое меню подразделяются на отдельные части: редактирование текста, фоновой цвет, размер, положение и смещение элементов, интенсивность теней и т.д.

Создание элементов упрощается тем, что пользователю не нужно прописывать CSS-код. Приложение позволяет импортировать исходный CSS-

код элементов и редактировать его. Также приложение взаимодействует с системой контроля версий (git), которая позволяет разработчику отправлять исходный код на сервер и синхронизировать его с рабочим проектом. Созданный элемент с кодом можно сохранить на компьютер при дальнейшем использовании. Как дополнение в приложении встроена галерея, где уже сформированы элементы на разный дизайн со своим исходным кодом. Также пользователь может зарегистрироваться в системе приложения и в любой момент времени обратиться к созданным элементам.

На рисунке 7 изображён интерфейс программного обеспечения.

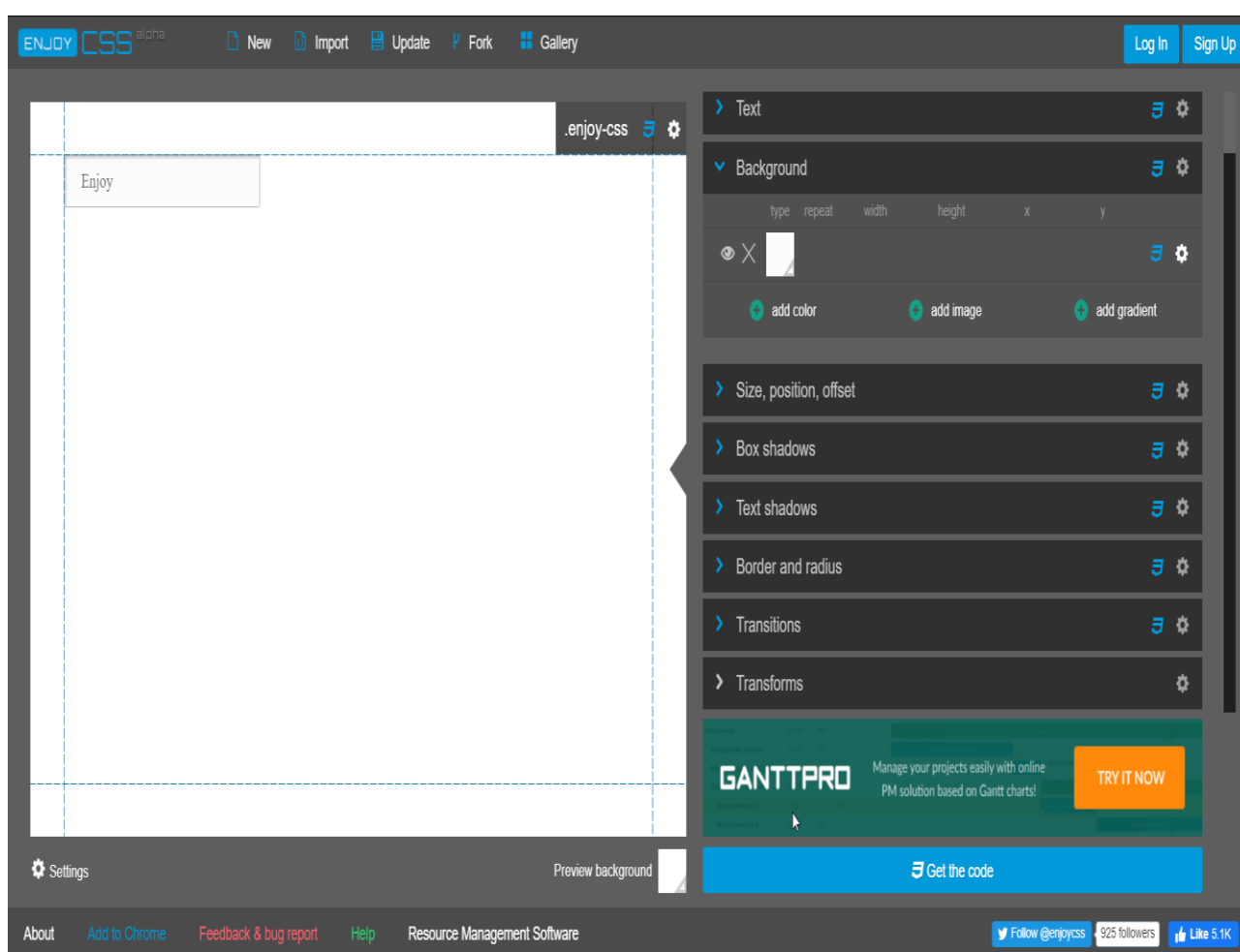


Рисунок 7 – Интерфейс программы Enjoy CSS

Однако приложение позволяет работать только с определёнными элементами для Web-сайта, что является существенным ограничением. Несмотря на то, что работа приложения направлена на поиск новых дизайнерских решений, оно не основано на эволюционных методах поиска. Формирование элементов носит характер ручного ввода.

### Основные плюсы системы:

- работа в окне браузера;
- удобный и понятный интерфейс;
- регистрация пользователя в системе;
- сохранение полученного результата;
- привязка к другим системам.

### Основные минусы системы

- создание элементов носит характер ручного управления;
- отсутствует механизм эволюционного поиска новых решений.

2. CSS Generator Tool очень схож по своей структуре с первым приложением (рисунок 8). Оно также работает через браузер и также предоставляет пользователю за короткое время создать элементы для Web-страниц. В большинстве своём эти отличия носят незначительный характер и не делают приложение хуже. Однако эта программа имеет ряд существенных дополнений, которые отсутствуют в первом приложении.

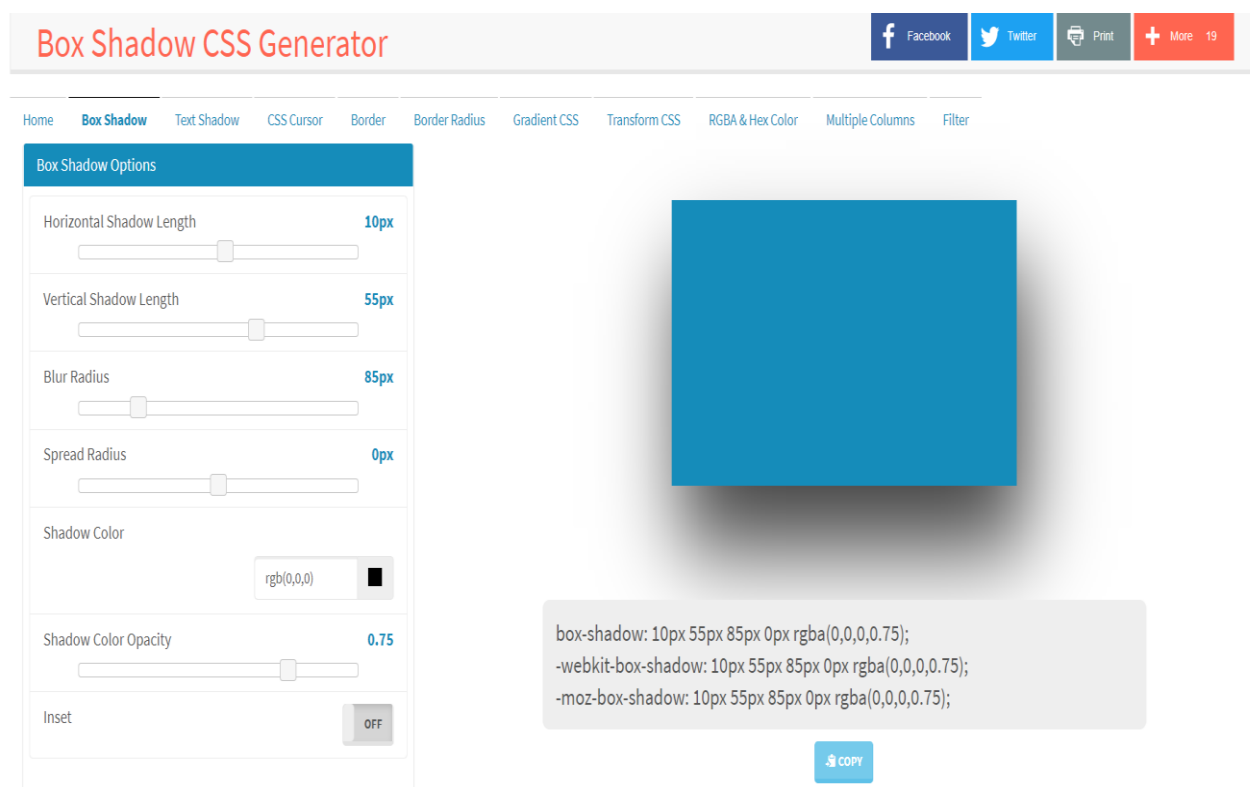


Рисунок 8 – Интерфейс программы CSS Generator Tool

Главный интерфейс приложения состоит из: главного экрана, верхнего и бокового меню навигации. На верхнем меню можно выбрать элементы, которые пользователь будет преобразовывать. В отличие от первого приложения, в котором стили CSS формировались с помощью выпадающего списка, во втором примере формирование элементов происходит в облегчённой форме с помощью специальных ползунков.

Также существенное отличие является в том, что основной акцент работы приложения приходится на блочные элементы. Изменяются цвет блоков, рамки, тени, а также происходят различные изменения относительно позиционирования на странице. Все эти манипуляции происходят в удобной для пользователя форме. Отличием также является то, что приложение может генерировать фон и накладывать различные фильтры на изображения (рисунок 9).

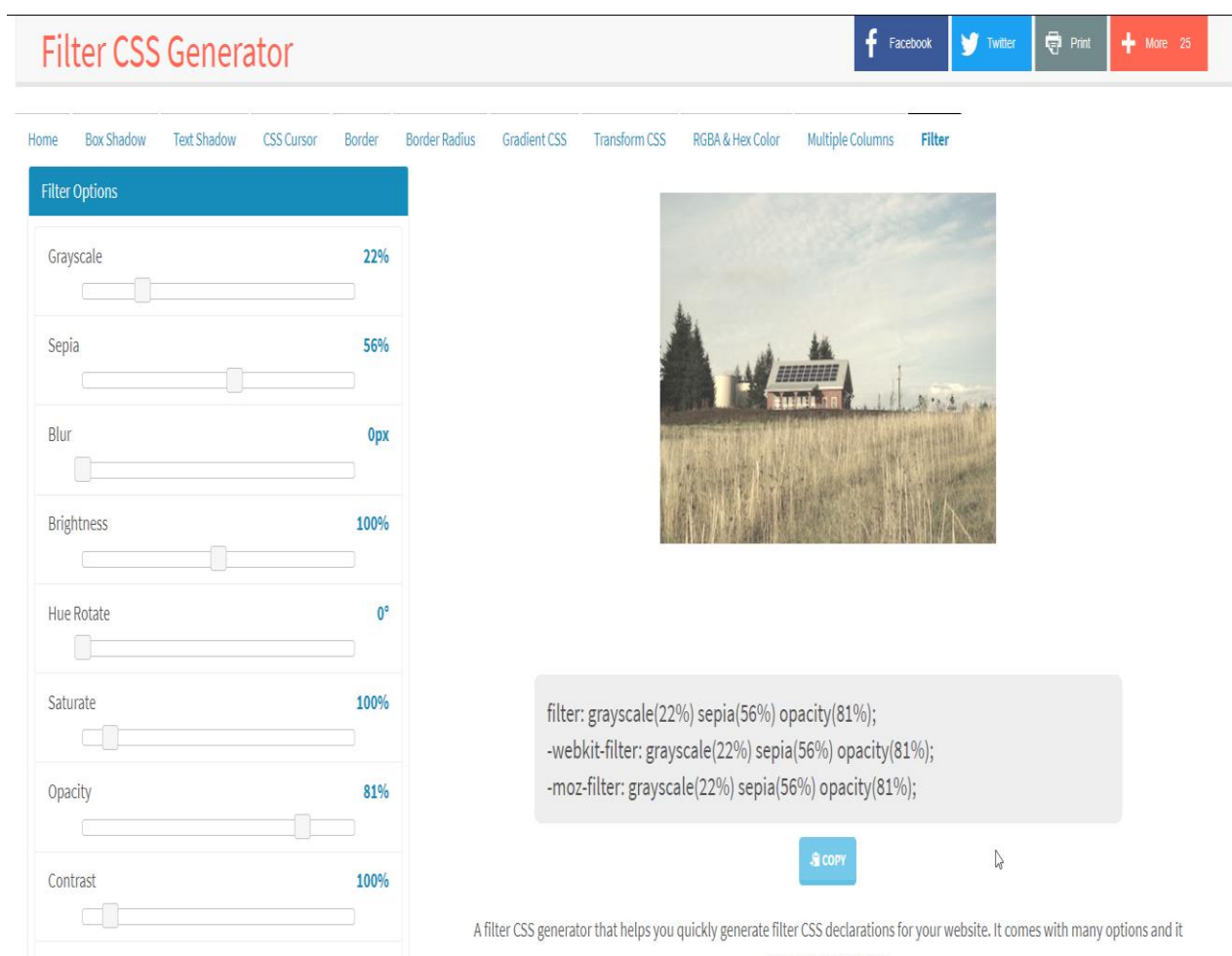


Рисунок 9 – Генерация фильтров для изображений

Что касается связи с другими программами, то приложение не взаимодействует с другими системами как это было в первом случае. Здесь

механизм действия несёт обратный характер. При формировании элементов автоматически генерируется CSS код и его можно скопировать себе на компьютер. Из отличий можно выделить тот факт, что приложение не требует регистрации и представляется в свободном доступе.

Также, как и в первом приложении все действия несут характер ручного управления со стороны пользователя и не основываются на работе эволюционных алгоритмов. Однако здесь прослеживается то, что CSS-код генерируется самостоятельно, то есть когда пользователь формирует стили, то эти стили «складываются» в CSS-код.

Основные плюсы системы:

- работа в окне браузера;
- удобный и понятный интерфейс;
- разнообразие в выборе элементов;
- изменение элементов генерируется в CSS-код.

Основные минусы системы:

- создание элементов носит характер ручного управления;
- отсутствует механизм эволюционного поиска новых решений;
- привязка к другим системам;
- регистрация пользователя в системе.

3. GradPad – это приложение, которое существенно отличается от двух других рассматриваемых программ. Работа программы направлена на формирование фонового изображения. Интерфейс приложения представляет собой экран с ползунками, которые настраивают градиент изображения. В верхней части экрана находится разделы с меню.

На рисунке 10 изображён интерфейс программного обеспечения.

Разработчики этого браузерного приложения упростили интерфейс до максимума. Пользователь не видит перед собой никаких всплывающих окон, никаких лишних пояснений. Верхнее меню состоит из пяти вкладок. Первая вкладка отвечает непосредственно за главный экран, на котором можно получить CSS-код.

Вторая вкладка отвечает за выбор цветовой гаммы, которую выбирает пользователь для генерации фона. Третья вкладка является переходом на так называемую страницу с опросом, где пользователь может оставить отзыв о том насколько хорошо работает программы по генерации фонового изображения. Также пользователь может написать предложение об улучшении работы программы. Как и в предыдущих приложениях пользователь имеет возможность не регистрироваться в данной программе.

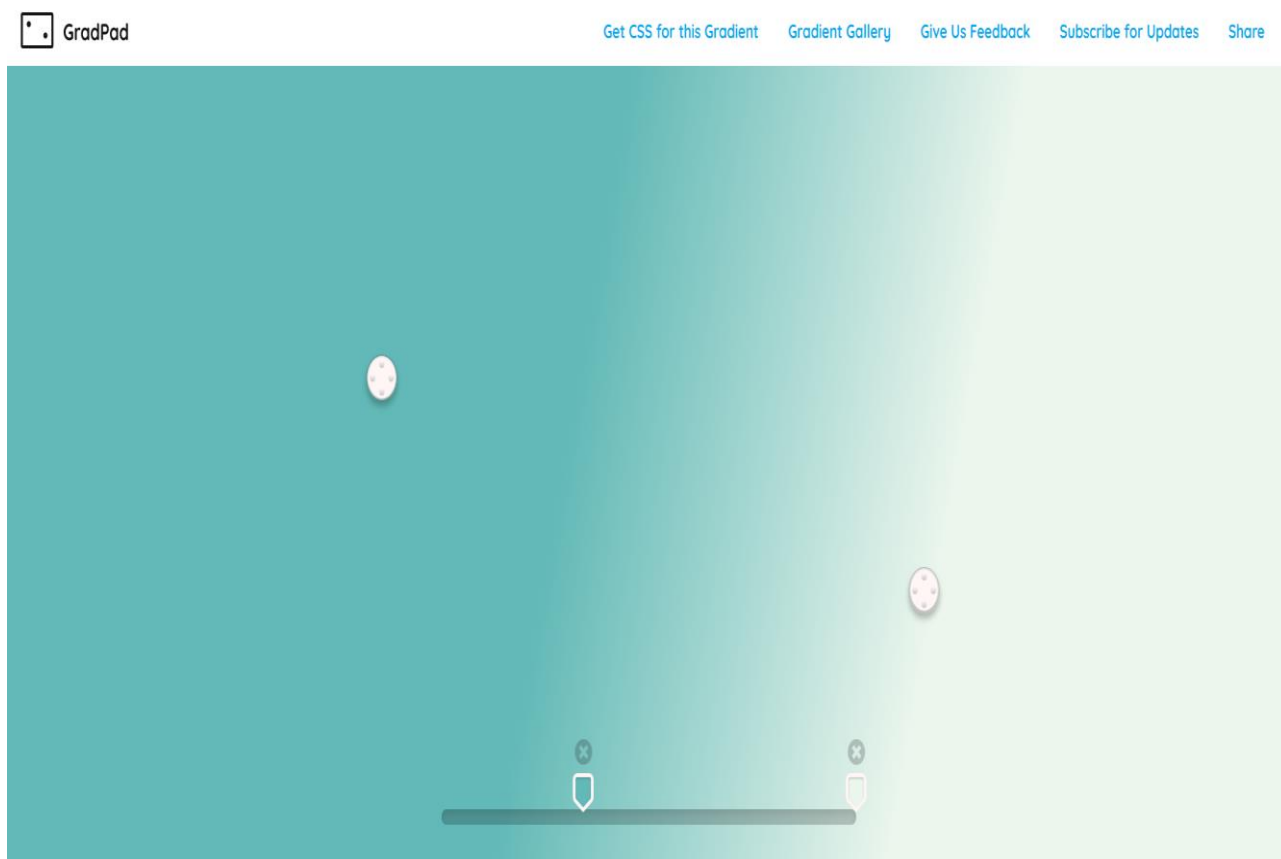


Рисунок 10 – Интерфейс программы GradPad

Основные плюсы системы:

- работа в окне браузера;
- удобный и понятный интерфейс;
- основной акцент идёт на генерацию фона;
- изменение элементов генерируется в CSS-код (рисунок 11).

Основные минусы системы:

- создание элементов носит характер ручного управления;
- отсутствует механизм эволюционного поиска новых решений;

- привязка к другим системам;
- регистрация пользователя в системе.

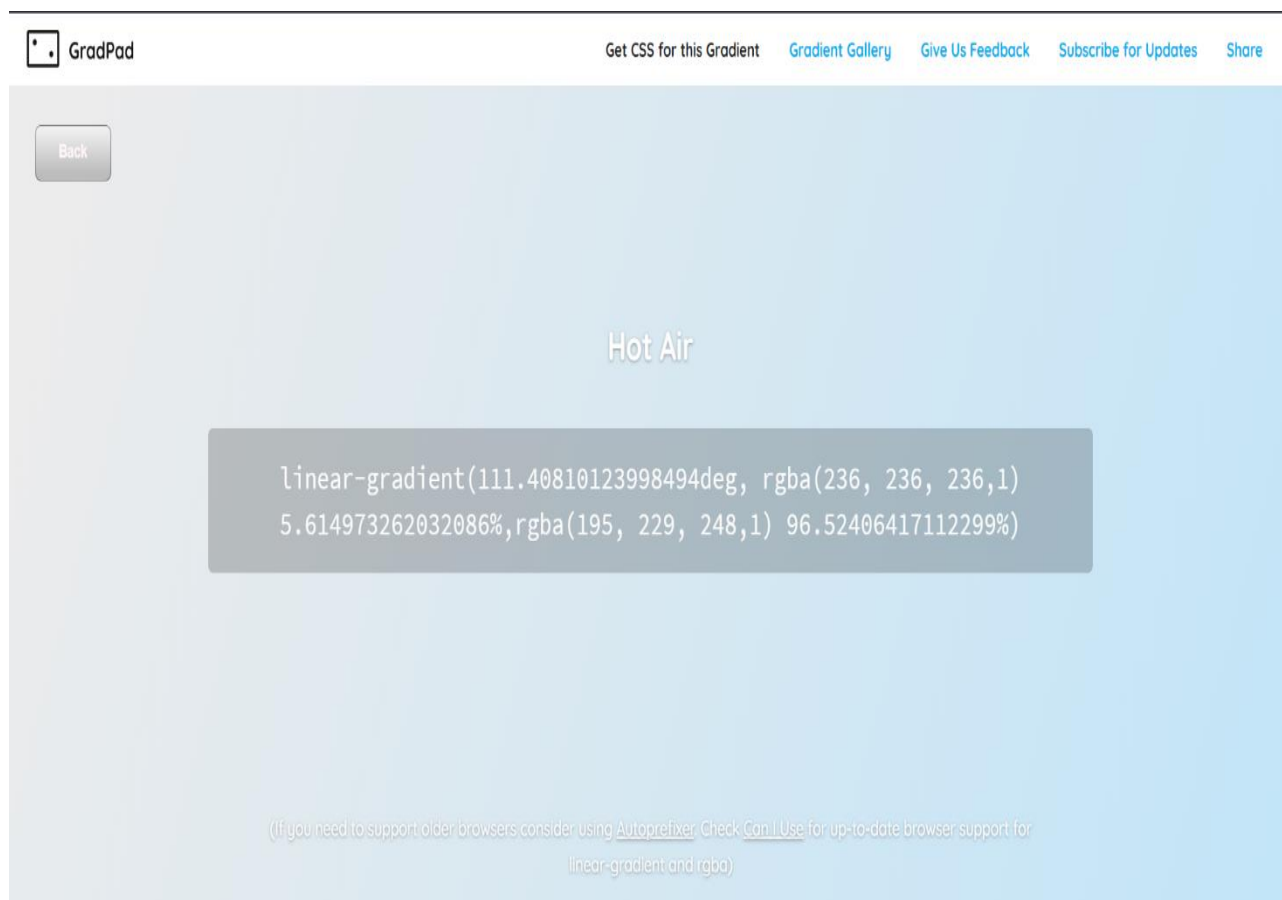


Рисунок 11 – Сгенерированный градиент

Несмотря на узконаправленную работу данного приложения, оно является одним из лучших генераторов фоновых изображений. Однако работа по генерированию изображений пользователь производит вручную, так же как в двух предыдущих рассмотренных программах.

4. Последний аналог стоит рассмотреть подробно, так как он очень приближённо отражает суть диссертационного исследования. Приложение называется Create template и базируется он на генерации Web-страниц для сайта. Генератор макетов, так же, как и предыдущие приложения является онлайн-сервисом, который генерирует макеты в архив, содержащий два файла: HTML и CSS, представляющие из себя базовую разметку (каркас) для Web-сайта.

На рисунке 12 изображён интерфейс приложения Create template.

Для формирования макета используется только блочная верстка (div-блоки). Шаблоны, которые генерируются в программе, протестированы и корректно отображаются в современных и популярных браузерах.

Приложение рассчитано на пользователей, которые имеют базовые навыки работы с языками гипертекстовой разметки. Однако тем, кто хочет познакомиться с Web-разработкой ближе этот сайт будет очень полезным.

### Создать html шаблон

Рисунок 12 – Интерфейс программы Create template

Теперь подробно разберём работу онлайн-приложения. Первое, что бросается в глаза — это меню, выполненное в стиле блочной модели. В первом блоке меню пользователь выбирает тип вёрстки HTML в которой будет генерироваться макет.

Рядом стоящий блок позволяет сбрасывать стили CSS. Это необходимо для того, чтобы браузер, который имеет по умолчанию свои встроенные отступы, обнулится. Таким образом пользователь подготовил всё необходимое для дальнейших настроек.

Далее идёт блок, который настраивает ширину макета. Пользователю предоставляется возможность выбрать два варианта настройки ширины:

фиксированная ширина и «резиновая» ширина. Это нужно для того, что разграничить статичный макет и адаптивный макет. Первый вариант делает ширину макета невосприимчивой к изменению экрана. Второй вариант является более универсальным и восприимчивым к изменению экрана.

Далее уже идут настройки, связанные с тем, как будут располагаться блоки внутри макета. Устанавливается ширина основных блоков: «шапки» сайта и «подвала» сайта. Они являются обязательными и без них не генерируется макет. Затем устанавливаются сайдбары – специальные блоки в которых отражается основная информация по сайту.

После выставления всех характеристик шаблона, пользователь может сгенерировать макет. На главном экране можно увидеть предварительный просмотр того как сгенерировался шаблон.

Для полного представления о результате работы алгоритма, пользователь может в отдельном окне увидеть то, как выглядит шаблон на всю ширину экрана браузера (рисунок 13). Также пользователь может скачать архив с шаблоном себе на компьютер.

Программа работает только с базовой настройкой разметки и не предполагает работы с дополнительными элементами в шаблоне. То есть приложение генерирует только базовый каркас страницы, а также устанавливает базовые стили. Также, как и другие программы, которые были рассмотрены, не работают на основе эволюционных алгоритмов, а имеют характер ручного управления.

Основные плюсы системы:

- работа в окне браузера;
- удобный и понятный интерфейс;
- генерация каркаса HTML;
- генерация базовых CSS-стилей;
- предварительный просмотр макета;
- скачивание архива с макетом.

Основные минусы системы:

- создание элементов носит характер ручного управления;

- отсутствует механизм эволюционного поиска новых решений;
- привязка к другим системам;
- регистрация пользователя в системе.

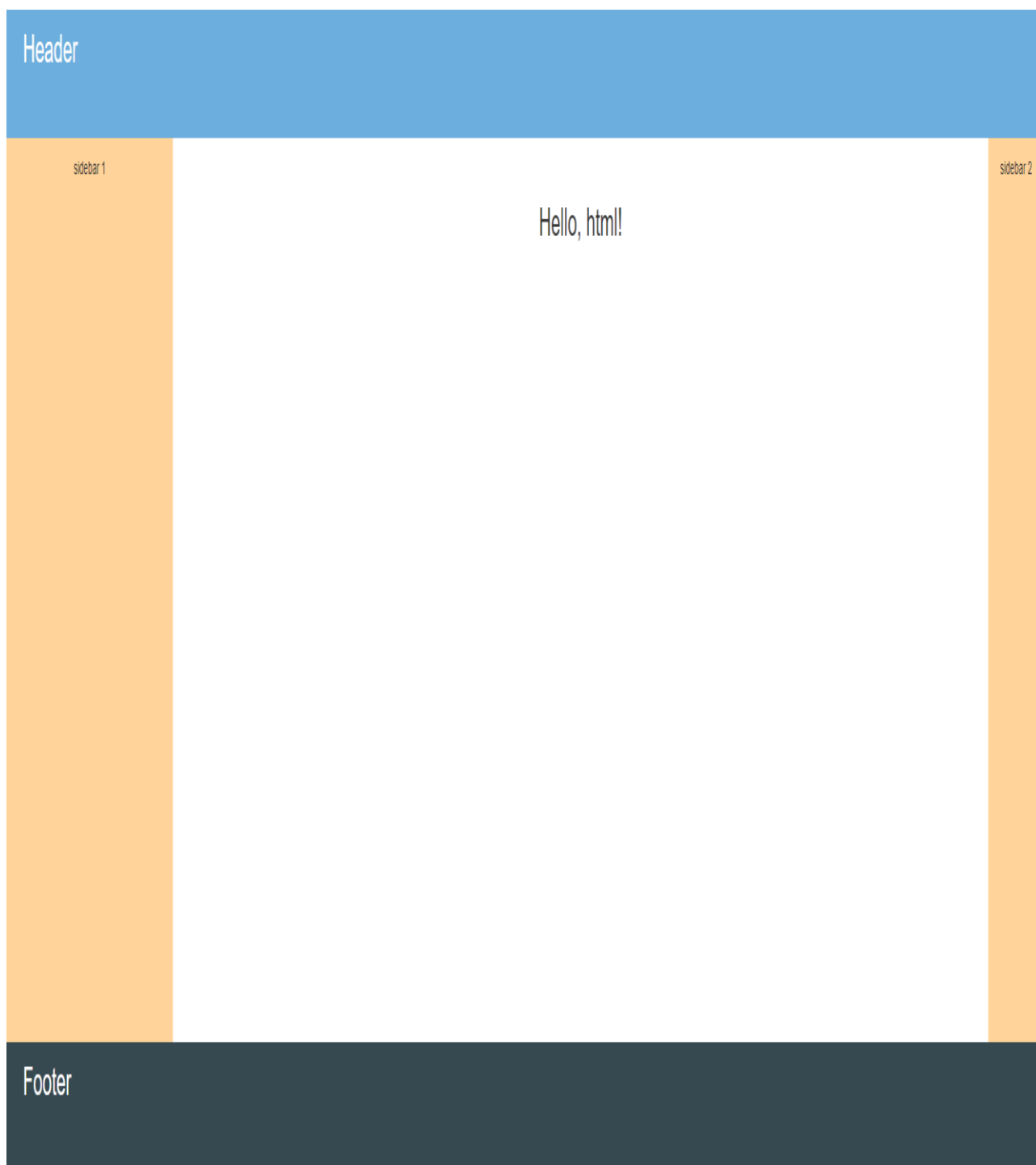


Рисунок 13 – Сгенерированный шаблон

Рассмотрев данные аналоги, можно выделить основные недостатки:

- 1) работа генераторов CSS акцентирует свою работу на отдельные элементы Web-страниц;

2) современные генераторы макетов Web-страниц работают исключительно через команды пользователя;

3) отсутствие у генераторов эволюционного подхода к генерированию новых оптимальных решений в нахождении Web-дизайна.

При проектировании программного обеспечения необходимо учесть преимущества представленных аналогов и устранить основные недостатки, связанные с эволюционным поиском.

Представленное программное обеспечение должно решить задачу, связанную с поиском оптимального дизайна для Web-сайта, с помощью генетических алгоритмов.

## **2.5 Описание информационной системы**

### **2.5.1 План описания информационной системы**

План описания по разработки информационной системы включает в себя следующую последовательность действий:

- a) описание системы и взаимодействие компонентов:
  - 1) описание структуры и назначений её частей;
  - 2) предназначения для использования информационной системы.
- b) функциональные требования к разрабатываемой системе:
  - 1) формирование схемы архитектуры информационной системы и описание связей подсистем;
  - 2) представление внешнего вида интерфейса системы в виде концептуальной модели процессов.
- c) указание сроков проектирования системы:
  - 1) предоставление перечня этапов разработки;
  - 2) определение сроков по ним.

При описании необходимо уделить внимание на следующие аспекты:

- разграничение прав и функций пользователей, работающих в приложении;
- представление алгоритмов, с помощью которых функционирует приложение;

- список компонентов, которые будут использоваться в разработке;
- описание взаимодействия пользователя с системой;
- описание входящих данных, с которыми будет работать программа;
- результат вывода данных на экран.

Переходим к описанию информационной системы, которое будет производить оптимальный поиск новых вариантов дизайна для Web-сайтов.

### **2.5.2 Взаимодействие компонентов системы**

Данное программное обеспечение классифицируется как абстрактная система и основана на выдвинутой гипотезе о нахождении оптимального дизайна для Web-страницы с помощью эволюционных алгоритмов.

Информационная система предназначена для поиска решений оптимизационных задач, связанных с генерацией дизайн-макетов для Web-сайта. Благодаря работе эволюционных алгоритмов пользователь получает различные варианты дизайнерских решений для исходной страницы макета.

Структура информационной системы имеет следующее представление – общий контейнер, куда помещены внешние и внутренние компоненты и инструменты управления ими.

Внешние компоненты, которые связаны с вводом данных и отображением их на главном экране состоят из:

- главного экрана приложения;
- интерфейса, с основными командами.

Командный интерфейс имеет следующие действия:

- загрузка макета;
- процесс генерации нового дизайна для всего макета;
- выбор для генерации отдельных элементов страницы;
- выбор адаптивного вида под различные устройства.

Внутренние компоненты программы – модули, основанные на алгоритмах, которые генерируют новый дизайн для макетов.

На рисунке 14 изображено взаимодействие внутренних и внешних компонентов системы.

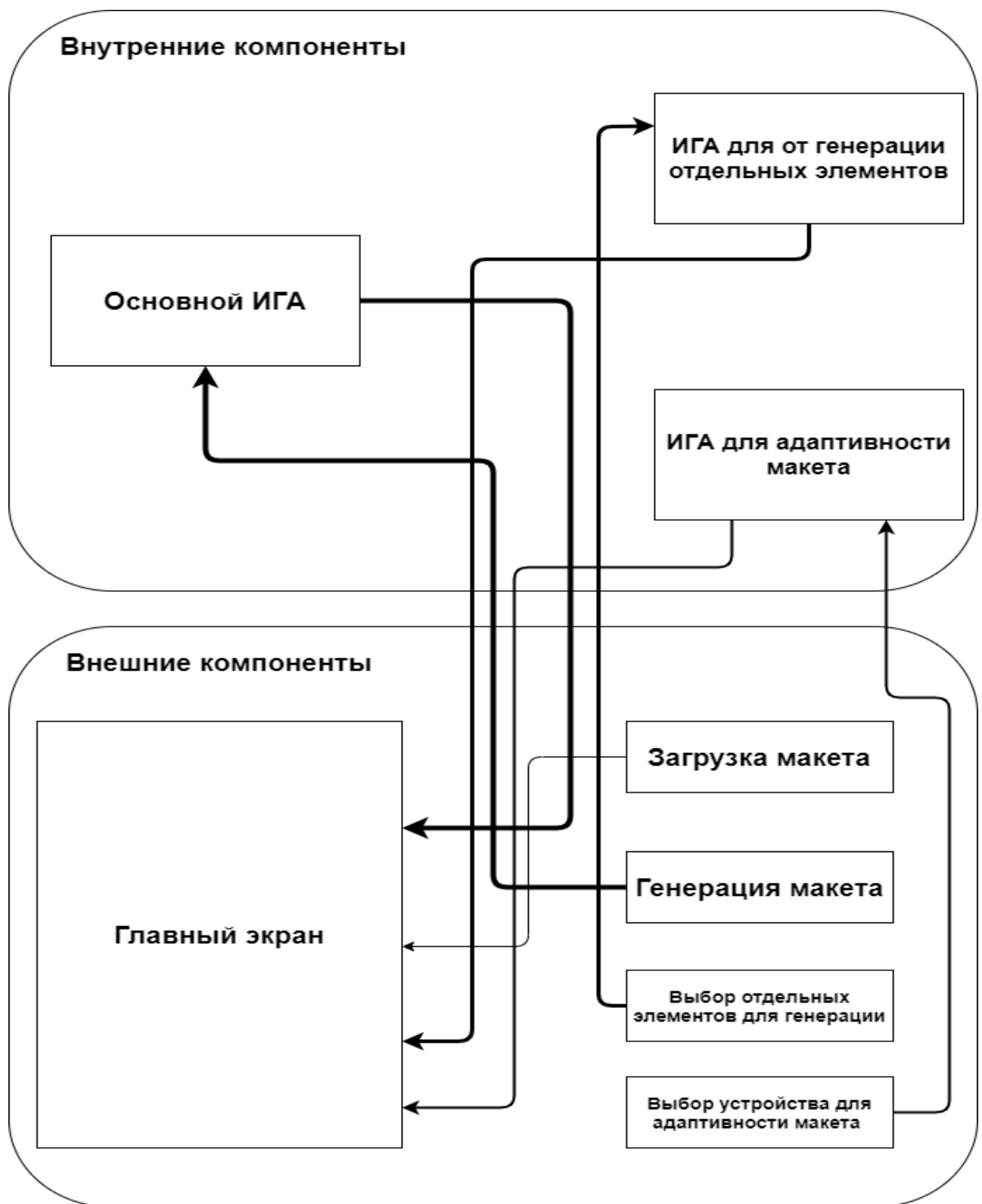


Рисунок 14 – Взаимодействие внутренних и внешних компонентов системы

В работе системы будут задействованы три интерактивных генетических алгоритма:

- алгоритм генерации нового дизайна страницы;

- алгоритм генерации отдельных элементов страницы, который изменяет свойства: блоков, кнопок и т.д.;
- генетический алгоритм, подстраивающий макет под другие устройства;

По представленной схеме взаимодействия компонентов сформируем схему архитектуры рассматриваемой информационной системы по генерации Web-интерфейсов (рисунок 15).

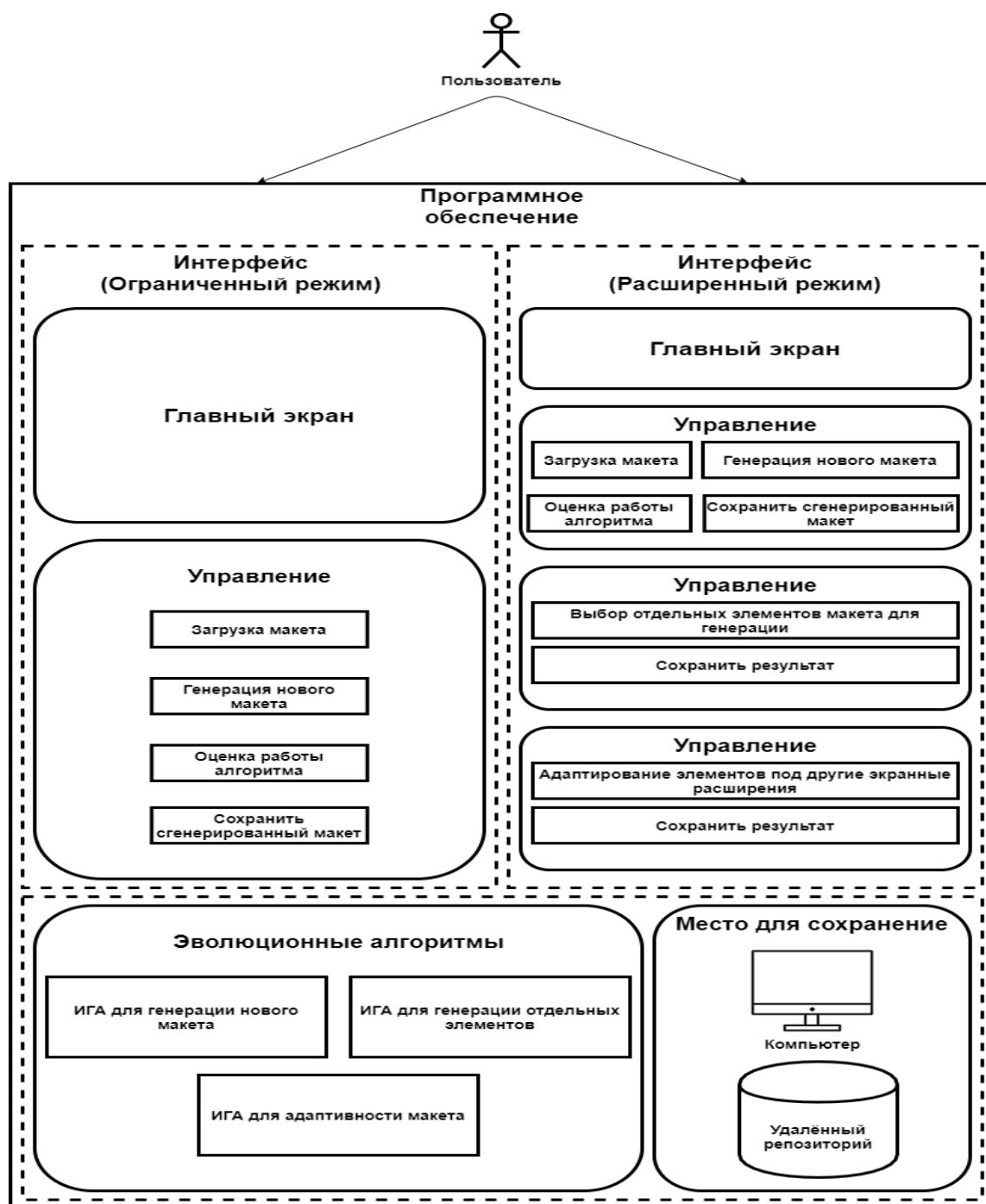


Рисунок 15 – Архитектура информационной системы

Представленная информационная система предназначена как для обычных пользователей, которые занимаются frontend-разработкой, так и для дизайнеров, которым нужно получить новые решения для создания внешнего вида макета. Каждый эволюционный алгоритм в информационной системе отвечает за определённые функции в системе, в зависимости от условия задачи.

### **2.5.3 Концептуальная модель системы и функциональные требования**

Главная цель пользователя, работающего с информационной системой – получить новые дизайнерские решения из исходного варианта макета. Результатом работы программного обеспечения является новый дизайн-макет, который был сгенерирован эволюционными алгоритмами из родительского макета.

Для понимания работы каждого элемента системы разрабатывается концептуальная модель. Концептуальная модель представляет собой диаграммы процесса, которые описывают функционирование внешних и внутренних компонентов приложения.

Диаграммы процесса построены на основе предложенной архитектуры системы. Информационная система по генерации Web-страниц работает в двух режимах.

Режим ограниченного функционала. Он представлен только для работы с формированием Web-интерфейса всего макета. В этом режиме пользователь имеет ряд существенных ограничений:

- генерация дизайн-макета имеет ограниченное число итераций;
- функция генерации отдельных элементов и функция генерации адаптивной версии может быть доступна после регистрации пользователя;
- сохранение результата может осуществляться только на компьютер.

На рисунке 16 показана контекстная диаграмма, которая описывает входные данные в ограниченном режиме.

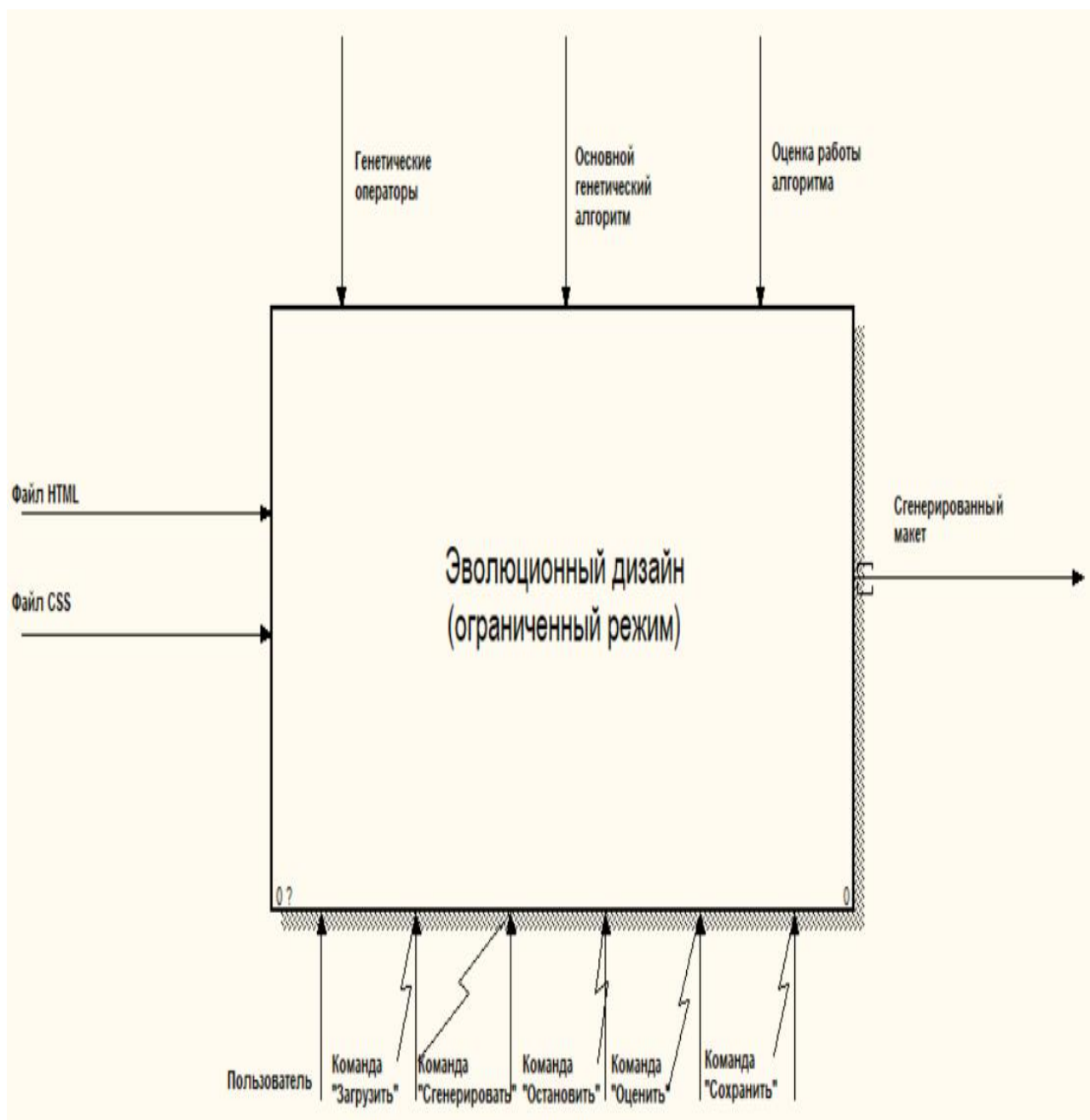


Рисунок 16 – Контекстная диаграмма входных данных в ограниченном режиме

Взаимодействие пользователя с системой в ограниченном режиме имеет следующую последовательность действий.

1. Пользователь загружает папку с проектом, в котором находится файл с расширениями HTML и CSS. Файл с исходным макетом отображается на главном экране браузера (рисунок 17).

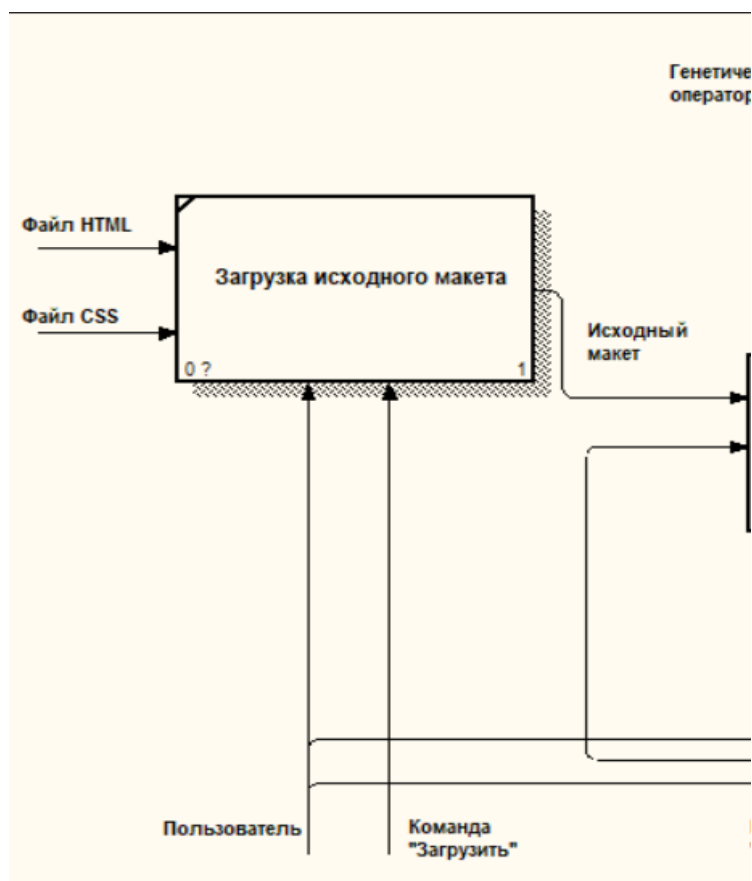


Рисунок 17 – Загрузка макета в систему

2. Пользователь запускает генерацию, чтобы получить новое дизайнерское решение по исходному макету (рисунок 18).

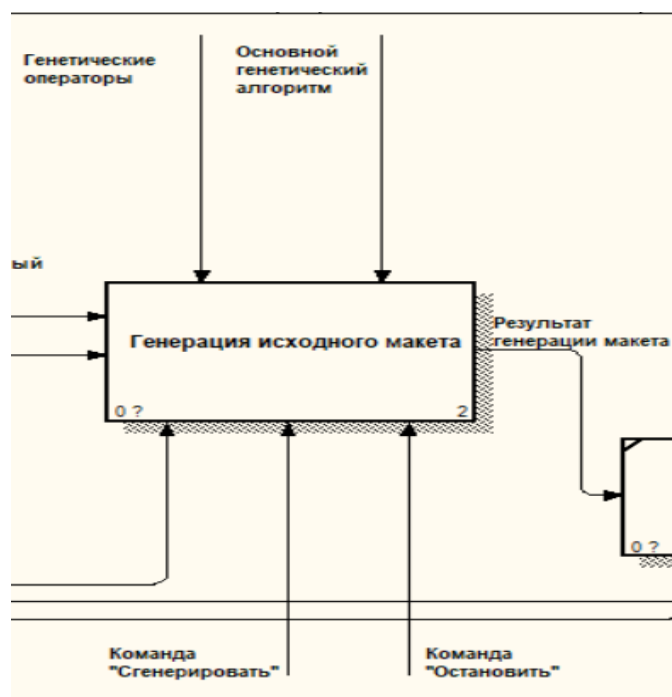


Рисунок 18 – Генерация макета

Эволюционный алгоритм (рисунок А.1) под действием генетических операций формирует макет-потомок из загруженного макета. Основные эволюционные операции происходят с атрибутами языка CSS.

3. Пользователь оценивает действие алгоритма и сохраняет новый макет страницы. Сохраняются сгенерированный макет в формате CSS. В ограниченной версии сохранение происходит на компьютер (рисунок 19).

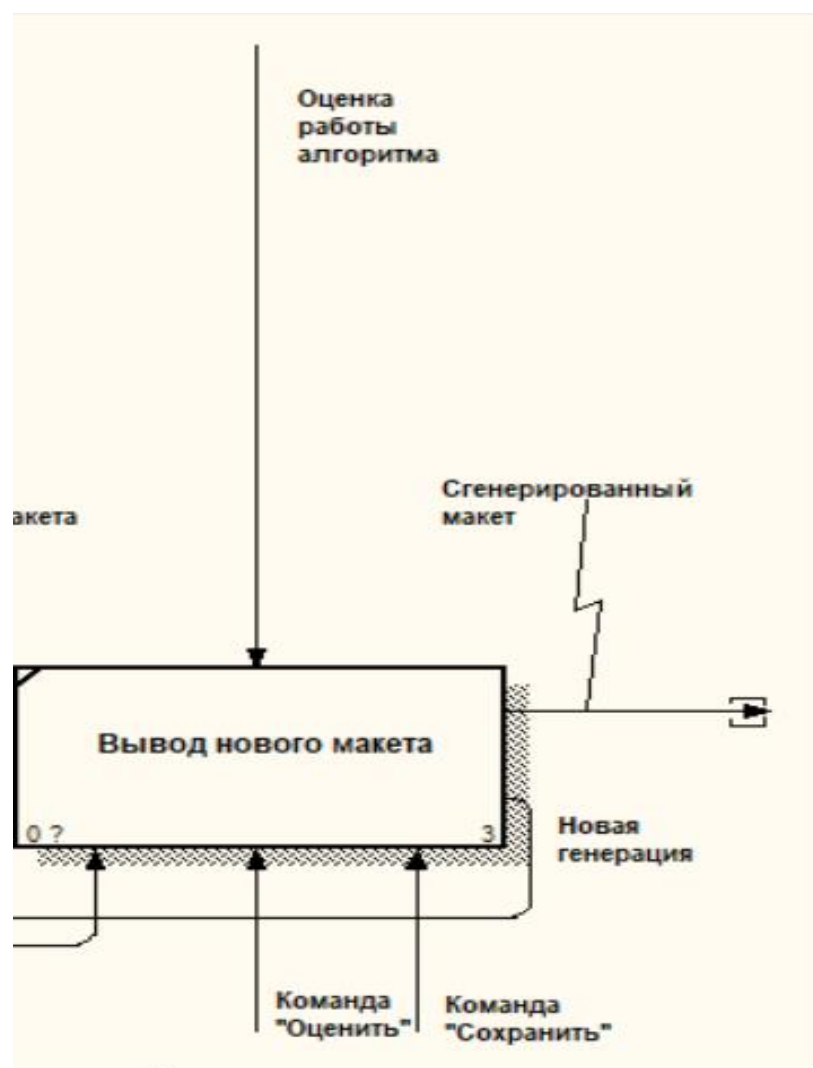


Рисунок 19 – Вывод и сохранение исходного макета

Характерной особенностью информационной системы является функция оценивания работы эволюционного алгоритма.

Оценивание происходит по пятибалльной системе. С помощью этой функции пользователь «посылает» сигнал в память программы тем самым сообщая о том, насколько хорошо алгоритм сгенерировал данный макет. Если пользователь оценил работу алгоритма на 1 - 3 балла, то при следующей

генерации алгоритм «учтёт» предпочтение пользователя и улучшит свою работу. Если пользователь оценил работу алгоритма на 4 - 5 баллов, то в следующий раз алгоритм будет «предлагать» именно такой вариант генерации макета.

Полная диаграмма декомпозиции представлена на рисунке 20.

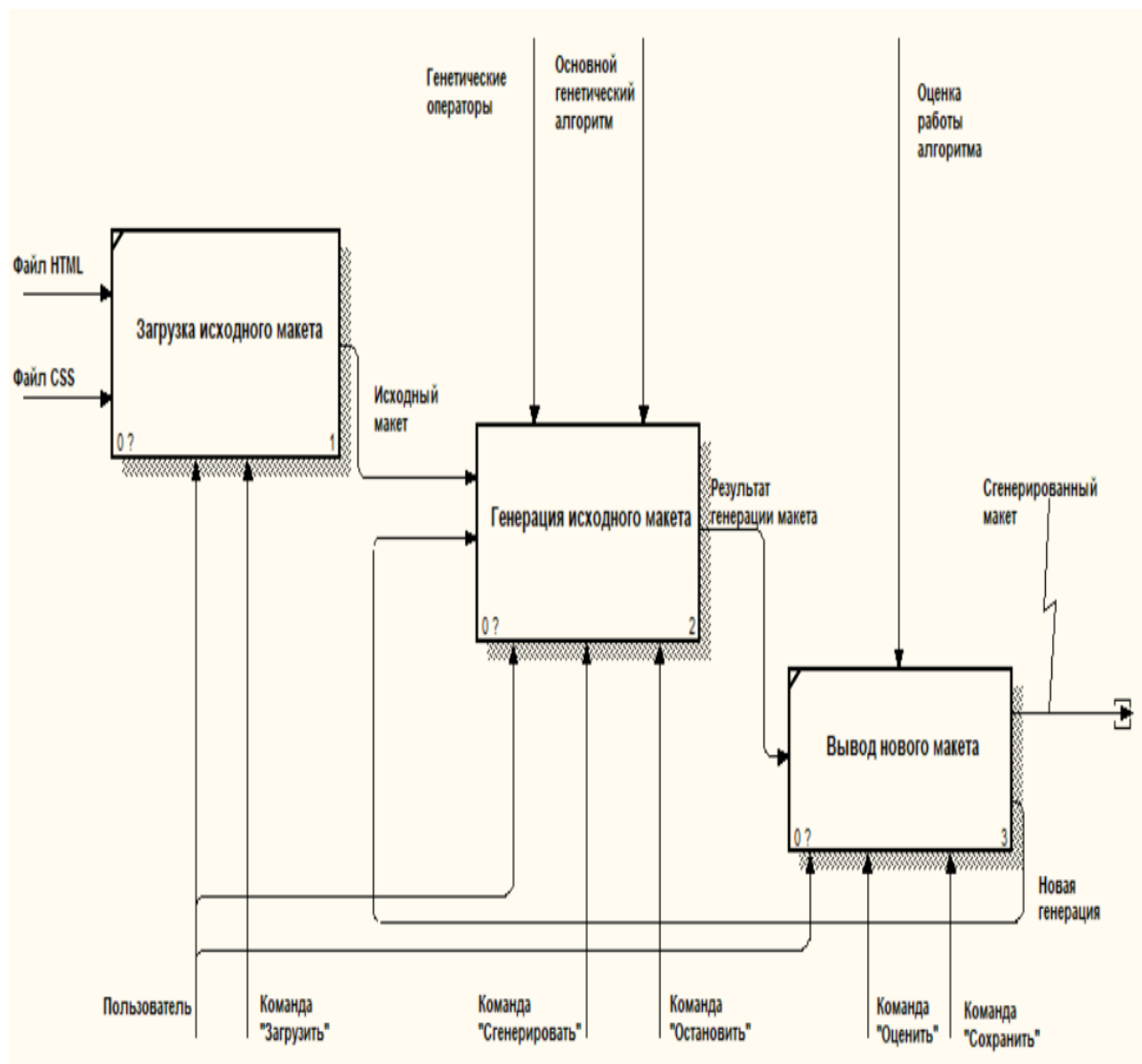


Рисунок 20 – Диаграмма декомпозиции работы системы в ограниченном режиме

Расширенный режим работы предполагает в себе дополнительные функции:

- неограниченное число итераций при генерации макета;
- возможность генерировать отдельные элементы исходного макета;

- возможность генерировать страницу, которая адаптируется под другие устройства;
- сохранение сгенерированного макета на локальный репозиторий, чтобы другие пользователи могли посмотреть результат работы эволюционных алгоритмов.

Контекстная диаграмма системы в расширенном режиме представлена на рисунке 21.

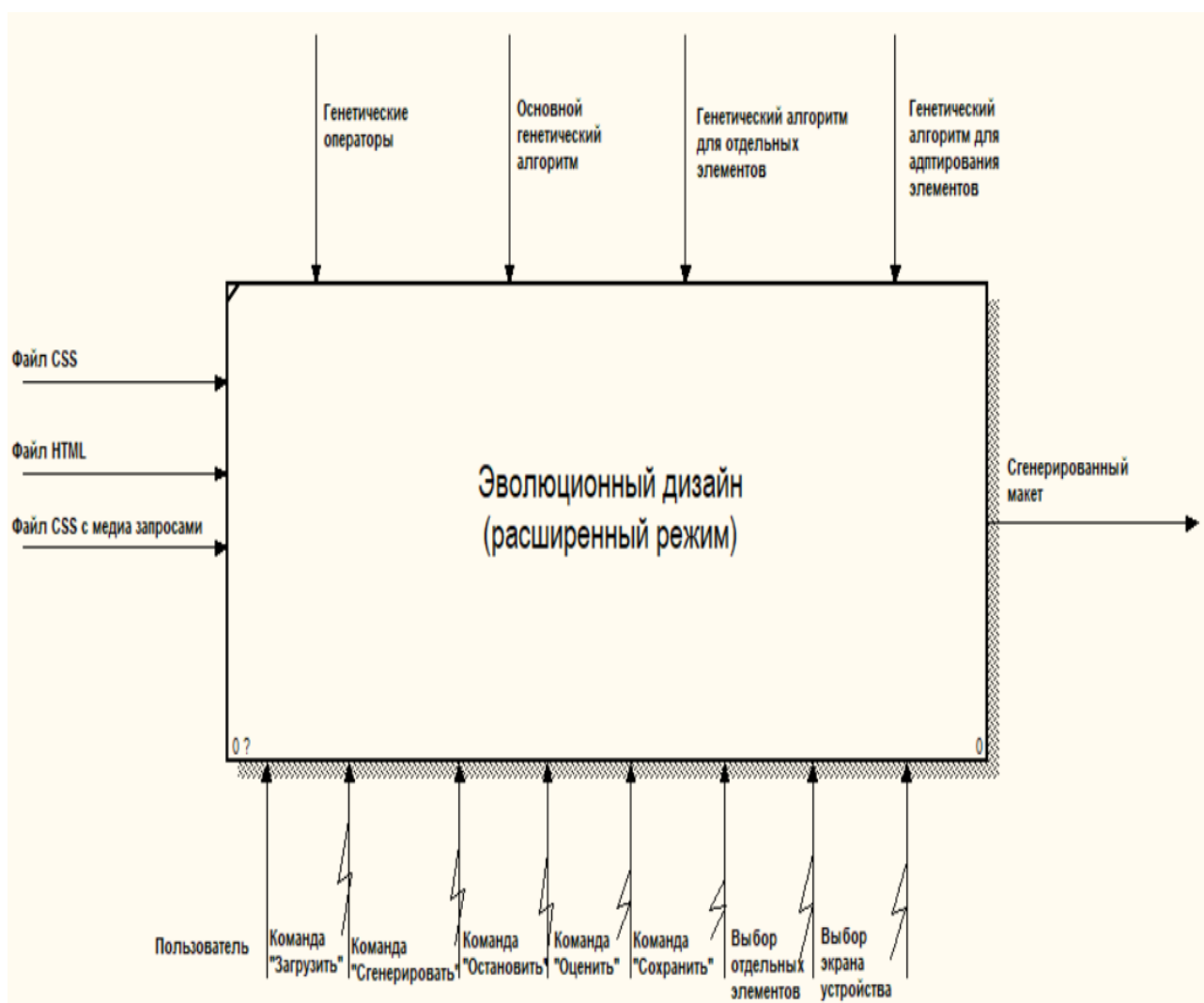


Рисунок 21 – Контекстная диаграмма входных данных в расширенном режиме

В расширенном режиме пользователю открывается весь функционал системы. Появляется возможность изменять под действием генетического алгоритма значение стилей CSS.

Дополнительные действия позволяют пользователю акцентировать внимание на генерацию некоторых свойств блочных элементов (рисунок 22).

Например, необходимо, чтобы блочный элемент имел: определённый цвет, толщину рамки и интенсивность тени. Пользователь находит и отмечает необходимые для генерации значения стилей.

Далее второй генетический алгоритм (рисунок А.2) генерирует между собой выбранные стили. Если сгенерированные стили блочных элементов удовлетворяют условия пользователя, то результат сохраняется на компьютер. Также сохранить результат можно в репозитории на локальном хостинге.

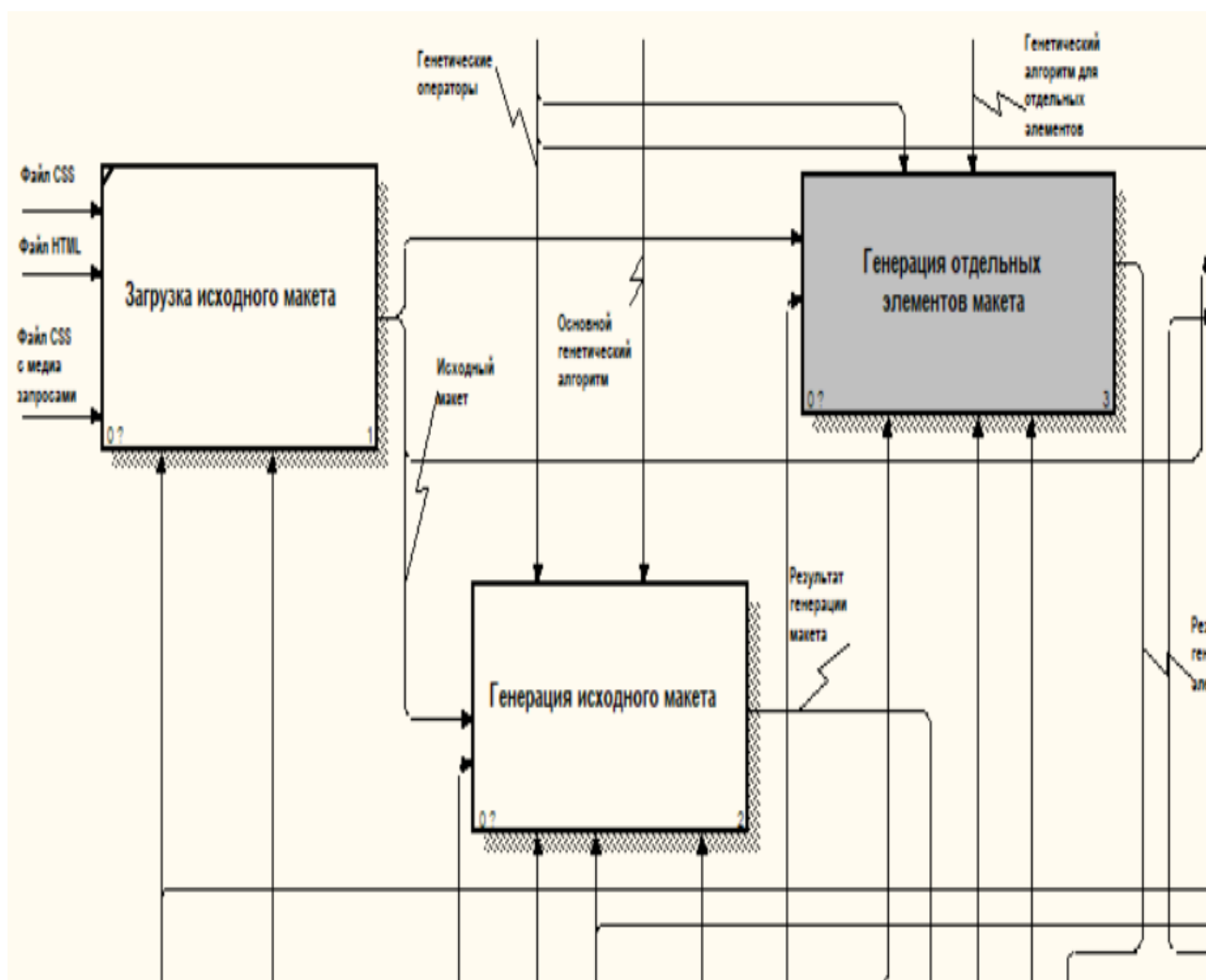


Рисунок 22 – Генерация дополнительных элементов макета

Дополнительной функцией системы является возможность пользователю адаптировать макет под разные расширения экрана (рисунок 23). С точки зрения Web-разработки адаптивирования Web-интерфейсов долгий процесс, требующий творческого подхода и нестандартного мышления. Данный процесс под действием эволюционного подхода предполагает уменьшить время на настройку

Web-интерфейса под различные устройства за счёт многообразия генерированных решений.

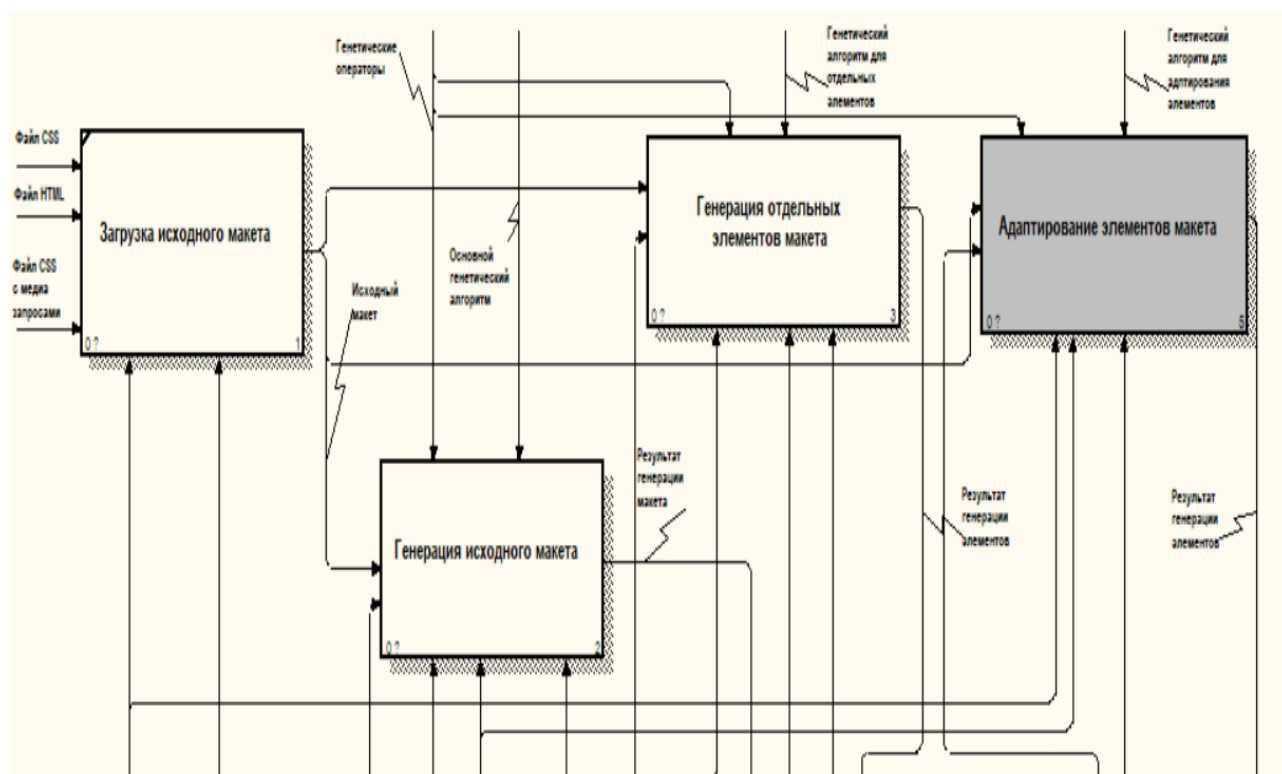


Рисунок 23 – Адаптивное элементов макета

В файле CSS присутствуют специальные правила – медиа запросы. Они позволяют управлять стилями элементов в зависимости от значений технических параметров устройств (рисунок 24).



Рисунок 24 – Загрузка макета с файлом медиа запросов

Это даёт возможность элементам Web-страницы подстраиваться различные диапазоны экранного расширения. В интерфейсе программы будут находиться переключатели, которые разделяют главный экран на две дополнительные части. Одна часть предназначена для расширения экранов планшета. Другая будет предназначена для экранов мобильных устройств. Выполнение третьего генетического алгоритма (рисунок А.3) происходит в файле с медиа запросами. В зависимости от того какой выбран диапазон экрана, алгоритм будет подстраивать под него элементы макета.

Полная диаграмма декомпозиции работы системы в расширенном режиме представлена на рисунке 25.

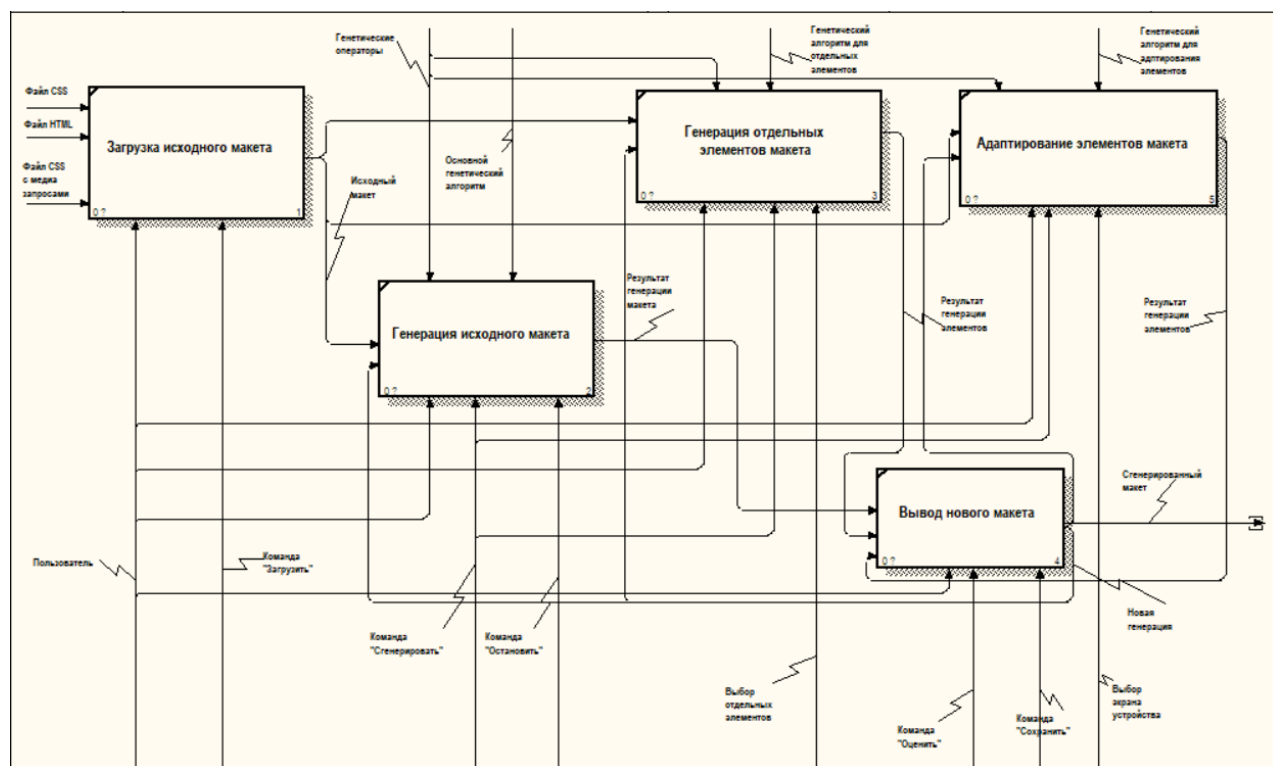


Рисунок 25 – Диаграмма декомпозиции работы системы в расширенном режиме

Разработка информационной системы с помощью эволюционных методов требует тщательной подготовки, поэтому нужно учесть различные нюансы на каждом этапе проектирования. В результате работа команд программы должна идти по чётко заданным алгоритмам.

#### 2.5.4 Стадии разработки информационной системы

Описание функциональных требований на примере концептуальной модели позволило сформировать основные стадии разработки и внедрения данной информационной системы. Основные стадии разработки данного программного обеспечения включают в себя следующие процессы.

1. Организационная структура проекта. Она согласуется на уровне начальника отдела проектировки и предназначена для организации взаимодействия между командами, вовлеченными в проект [16].

2. Сбор требований и разработка технического задания. Участники проекта определены, и на этом этапе собираются и утверждаются требования к будущей информационной системе [17].

3. Эскизный проект. Эскизный проект разрабатывают с целью установления принципиальных (конструктивных, схемных и др.) решений изделия, дающих общее представление о принципе работы и (или) устройстве изделия, когда это целесообразно сделать до разработки технического проекта или рабочей документации [18].

4. Технический проект. Работой, выполняемая на этой стадии, является разработка основных технических решений по создаваемой системе и окончательное определение ее сметной стоимости [19].

5. Рабочая документация. Работой, на этой стадии, является выпуск рабочей документации на создаваемую систему. Этап завершается выпуском рабочего проекта ИС, состоящего: из проектной документации, необходимой и достаточной для приобретения, монтажа и наладки комплекса технических средств системы; документации программного и организационного обеспечения, необходимого и достаточного для наладки и эксплуатации системы; программ специального программного обеспечения на машинных носителях [19].

6. Ввод в эксплуатацию. Этап внедрения информационной системы включает ввод в опытную эксплуатацию технических и программных средств,

проведение опытной эксплуатации всех компонентов и системы в целом, сдачу в эксплуатацию и подписание актов приемки-сдачи работ [20].

7. Дальнейшее сопровождение. Завершающая стадия представляет собой сопровождение информационной системы на всём этапе его жизненного цикла [21].

В таблице 1 приведено подробное описание стадий для разрабатываемой информационной системы.

Таблица 1 – Стадии разработки информационной системы

| Стадии                                            | Этапы работ                             | Результат работы                                              | Срок выполнения |
|---------------------------------------------------|-----------------------------------------|---------------------------------------------------------------|-----------------|
| Организационная структура                         | Заключение трудового договора           | Договор заключён                                              | 8 дней          |
|                                                   | Согласование процедур управления        | Процедуры управления проектом и устав проекта согласованы     | 5 дней          |
|                                                   | Формирование команды                    | Команда проекта сформирована                                  | 5 дней          |
| Сбор требований и разработка технического задания | Разработка концептуальной модели данных | Концептуальная модель данных                                  | 10 дней         |
|                                                   | Разработка технического задания         | Разработано техническое задание и различные приложения к нему | 15 дней         |
|                                                   | Согласование и утверждение              | Согласованное и утвержденное техническое задание              | 10 дней         |

Продолжение таблицы 1

| Стадии          | Этапы работ                                                              | Результат работы                                                                                    | Срок выполнения |
|-----------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------|
| Эскизный проект | Обоснование выбора программного обеспечения и технической инфраструктуры | Программное обеспечение выбрано. Утвержден план формирования необходимой технической инфраструктуры | 10 дней         |
|                 | Эскизное проектирование процессов ETL                                    | Общее описание процессов извлечения данных, алгоритмов трансформации, загрузки и агрегации данных   | 8 дней          |
|                 | Проектирование интерфейса пользователя                                   | Общее описание интерфейса ввода и предоставления данных                                             | 6 дней          |
|                 | Согласование и утверждение                                               | Согласованная и утвержденная пояснительная записка к эскизному проекту                              | 5 дней          |

Продолжение таблицы 1

| Стадии             | Этапы работ                                         | Результат работы                                                                                                                    | Срок выполнения |
|--------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Технический проект | Определение функциональной и технической архитектур | Описание функциональной архитектуры<br>Описание технической архитектуры                                                             | 15 дней         |
|                    | Разработка физической модели данных                 | Физическая модель данных - описание таблиц БД, индексов, секций и других объектов БД                                                | 10 дней         |
|                    | Проектирование процессов ETL                        | Описание процессов извлечения данных, алгоритмов трансформации и обеспечения качества данных, процессов загрузки и агрегации данных | 15 дней         |
|                    | Проектирование интерфейса пользователя              | Описание бизнес-слоя данных, интерфейсов ввода и предоставления данных, разграничения прав доступа                                  | 15 дней         |

Продолжение таблицы 1

| Стадии               | Этапы работ                                              | Результат работы                                                                                                                  | Срок выполнения |
|----------------------|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Технический проект   | Оформление технического проекта                          | Пояснительная записка к техническому проекту                                                                                      | 18 дней         |
|                      | Согласование и утверждение                               | Согласованная и утвержденная пояснительная записка к техническому проекту                                                         | 8 дней          |
| Рабочая документация | Разработка рабочей документации на систему и на её части | Готовая документация                                                                                                              | 25 дней         |
|                      | Согласование и утверждение                               | Согласованная и утвержденная рабочая документация                                                                                 | 10 дней         |
| Ввод в эксплуатацию  | Проведение опытной эксплуатации                          | Опытная эксплуатация проведена.<br>Устранение неисправностей, доработка ПО и дополнительная наладка технических средств проведены | 10-20 дней      |

Окончание таблицы 1

| Стадии                   | Этапы работ                                      | Результат работы                                          | Срок выполнения |
|--------------------------|--------------------------------------------------|-----------------------------------------------------------|-----------------|
| Ввод в эксплуатацию      | Завершение работ по вводу в эксплуатацию         | Подписание документов об окончании работ                  | 3 дня           |
| Дальнейшее сопровождение | Обновление системы в случае добавления изменений | Устранение неполадок и улучшение программного обеспечения | 365 дней        |

Представленная таблица является приближённым сценарием по проектированию информационной системы. В рамках научной исследовательской работы спроектировать данную информационную систему затруднительно.

### **Выводы**

Во второй главе научно-исследовательской работы были описаны этапы разработки программного обеспечения эволюционными методами.

В данном разделе была выдвинута основная гипотеза предмета исследования, которая объясняет эффективность использования генетических алгоритмов в языке каскадных таблиц стилей.

На основании гипотезы выстраиваются следующие схемы взаимодействия:

- взаимодействие пользователя с ИГА;
- взаимодействие эволюционного алгоритма в языке CSS.

На примере этих схем выстраивается основная методика разработки информационной системы, которая находит новые решения, связанные с дизайном Web-страницы.

Приведены несколько аналогов, которые наиболее широко отражают процесс нахождения новых дизайнерских решений. Выявлены основные плюсы системы и минусы данных онлайн-приложений по созданию Web-интерфейсов.

Приведён подробный план описания информационной системы по генерации дизайн-макетов. В нём присутствуют следующие этапы:

- a) описание взаимодействий внутренних и внешних компонентов системы;
- b) разработка концептуальной модели программного интерфейса информационной системы;
- c) описание функциональных требований для программного обеспечения:
  - 1) разграничение прав для пользователей;
  - 2) работа пользователей в разных режимах системы;
  - 3) описание входящих данных, с которыми будет работать программа;
  - 4) конечный вывод данных на экран.

На основе данной концептуальной модели выдвинуты основные стадии разработки системы и установлены сроки по ним.

В заключительном разделе научно-исследовательской работы выделяется результат практической значимости данного исследования. Он заключается в подтверждении необходимости использования метода эволюционного поиска для нахождения новых дизайнерских решений:

- показать работу трёх генетических алгоритмов посредством концептуальных диаграмм;
- показать дизайн-макеты, созданные на основе работы эволюционных алгоритмов;
- провести соответствующий вывод.

### 3 Анализ работы эволюционных алгоритмов

#### 3.1 Концептуальные модели работы генетических алгоритмов

Рассмотрим внутреннюю структуру процессов эволюционной генерации. По условию описания информационная система имеет три генетических алгоритма:

- ГА для всего дизайн-макета;
- ГА для отдельных элементов;
- ГА для адаптирования элементов макета под другие размеры экрана.

Для рассмотрения процессов проведём декомпозицию следующих диаграмм:

- диаграмма генерации исходного макета (смотри рисунок 18);
- диаграмма генерации отдельных элементов макета (смотри рисунок 22);
- диаграмма генерации элементов макета под другие экранные расширения (смотри рисунок 23).

Для построения декомпозиции воспользуемся диаграммами вида IDEF3. Этот вид диаграмм позволит иметь наглядное представление о процессе генерации новых дизайнерских решений.

Отличительной особенностью диаграмм IDEF3 является специальные объекты «перекрёстки» (рисунок 26).

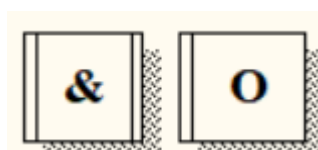


Рисунок 26 – Синхронный (логическое И) перекрёсток и асинхронный перекрёсток (логическое ИЛИ)

Они схожи по своим функциональным требованиям с логическими операторами и предназначены для отображения логики взаимодействия стрелок (потоков) при слиянии и разветвлении. «Перекрёстки» бывают синхронными и асинхронными. Это позволяет понять какие процессы работы алгоритма

выполняются одновременно, а какие нет. На основе блок-схем генетических алгоритмов (приложение А) построим концептуальную модель работы генетических алгоритмов (рисунок 27).

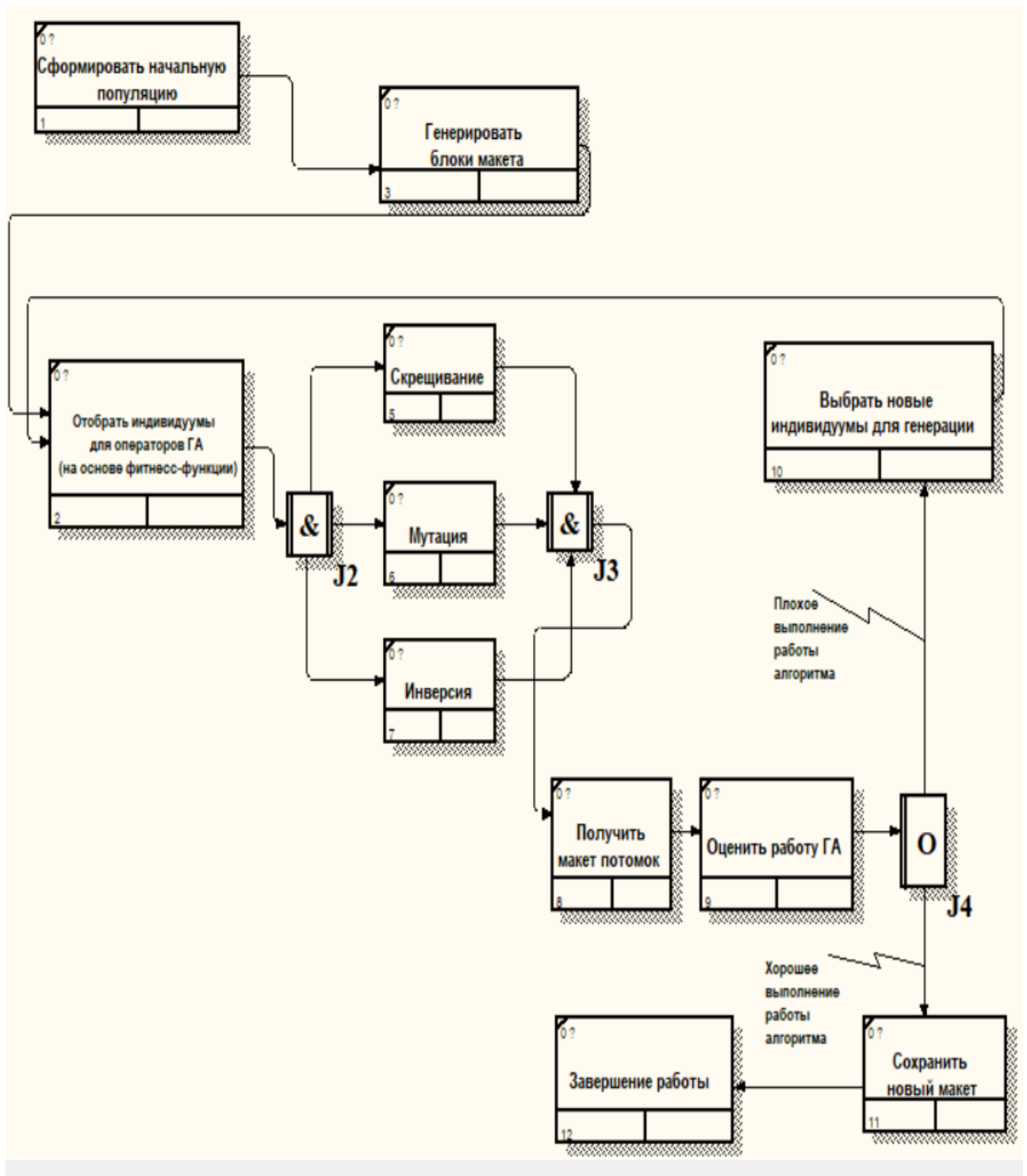


Рисунок 26 – Декомпозиция диаграммы генерации дизайн-макета

Следует отметить, что процессы работы генетических операторов выполняются одновременно, так как они связаны синхронным перекрёстком.

При этом один или все предшествующие процессы должны быть завершены. Там, где происходит оценка работы алгоритма установлен асинхронный перекрёсток (логическое ИЛИ), которое позволяет алгоритму (если пользователь оценил работу на высокий балл) перейти к сохранению дизайн-макета. Если пользователь поставил низкую оценку работе алгоритма, то алгоритм переходит на этап нового отбора индивидуумов для генерации. Результат работы эволюционного алгоритма приведён на рисунке Б.1. Следующая декомпозиция для диаграммы генерации отдельных элементов представлена на рисунке 28.

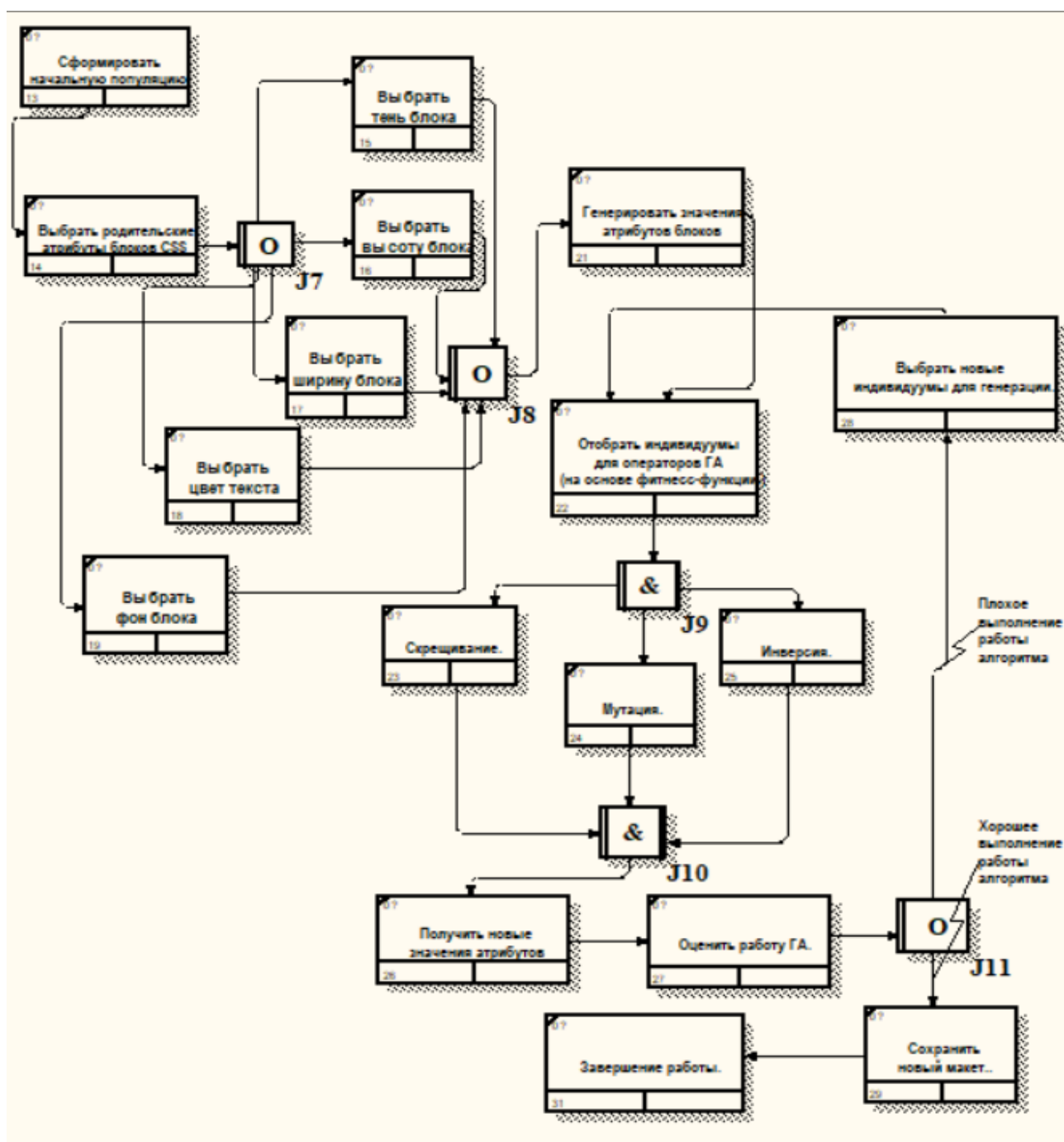


Рисунок 28 – Декомпозиция диаграммы генерации отдельных элементов макета

Работа второго алгоритма направлена на генерацию не всего макета, а только отдельных его элементов: кнопок, ссылок, текста и т.д. Для этого пользователь в информационной системе выбирает один из атрибутов блока: фон, тень, высоту и т.д. и посылает алгоритму сигнал о генерации значений этих атрибутов. Выбор атрибутов в декомпозиции происходит через асинхронный перекрёсток. Результат работы эволюционного алгоритма приведён на рисунке Б.2. Заключительная декомпозиция представлена для генерации элементов макета под другие экранные расширения (рисунок 29).

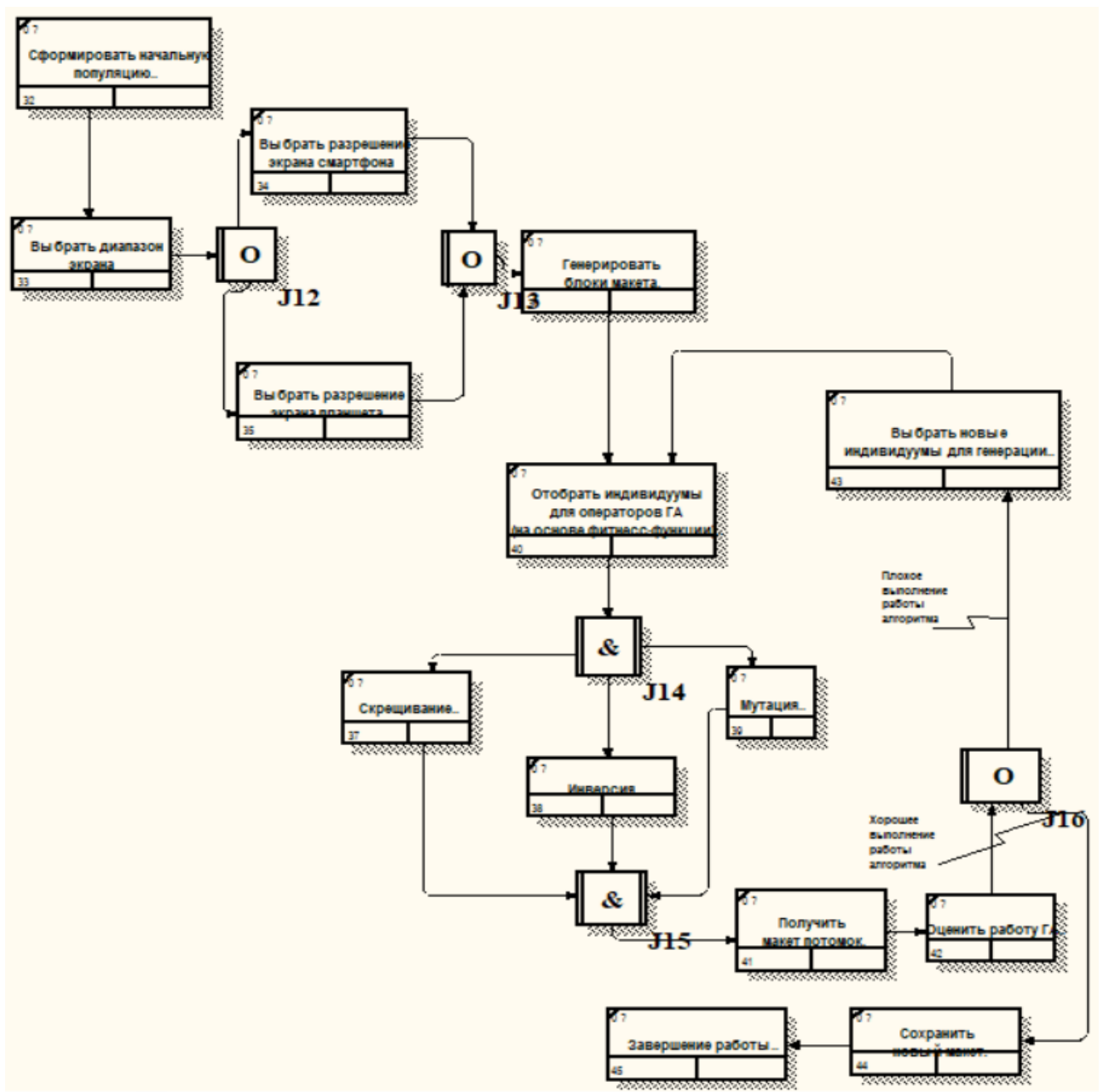


Рисунок 29 – Декомпозиция диаграммы генерации отдельных элементов макета

Третий алгоритм ориентирован на то, чтобы подстроить элементы макета под экраны других устройств. По замыслу пользователь в информационной системе выбирает один из двух устройств: планшет или смартфон. Выбор происходит через асинхронный перекрёсток. В зависимости от выбранного устройства эволюционный алгоритм начинает генерировать блоки таким образом, чтобы максимально точно подстроить под необходимый экран. Результат работы эволюционного алгоритма приведён на рисунках Б.3 и Б.4.

### **Выводы**

Завершающая глава научной исследовательской работы посвящена описанию работы эволюционных алгоритмов в рассматриваемой информационной системе.

Для разработки концептуальных моделей работы алгоритмов были использованы диаграммы вида IDEF3. Они являются декомпозицией процессов, которые были показаны во второй главе научной исследовательской работы. Декомпозиция построена на основе блок-схем, которые описывают процессы работы эволюционных алгоритмов. Каждый генетический алгоритм выполняет свои определённые функции и каждый следующий шаг является детерминированным.

Результатом работы генетических алгоритмов является новые оптимальные варианты дизайнерских решений для Web-страниц.

## Заключение

В процессе написания научной исследовательской работы была рассмотрена важная тема – разработка дизайна Web-сайтов с помощью эволюционных методов. Одним из этих методов является генетические алгоритмы. Основной идеей эволюционных методов является получение конкретного оптимального решения в сложной задаче оптимизации. Задачей для исследования была выбрана область, связанная с разработкой дизайн-макетов. Результатом данного исследования было показать эффективность взаимодействия эволюционных алгоритмов с языком каскадных таблиц стилей CSS.

В ходе исследования были проанализированы основные направления применения эволюционных методов. Были приведены труды, связанные с применением эволюционных методик в сфере разработки Web-интерфейсов. На основе этих проанализированных исследований была выдвинута гипотеза, которая заключалась во внедрении генетического алгоритма в язык CSS. На основе этой гипотезы была разработана схема архитектуры информационной системы и создана концептуальная модель. Информационная система показана в виде диаграмм процессов, которые подробно отображают работу системы. Далее были представлены диаграммы процессов работы генетических алгоритмов внутри системы. В них подробно описывается каждый шаг данных алгоритмов. Результатом являются сгенерированные дизайн-макеты.

Подводя итог данного исследования хочется отметить, что задача по разработке хорошего дизайна для Web-страницы является трудоёмкой. Дизайнер тратит очень много времени на создание оптимального варианта дизайна-макета. Совместная работа эволюционных алгоритмов и каскадных таблиц стилей помогает решить проблему разработки Web-интерфейсов сайта, а также снижает трудоёмкость при поиске оптимальных вариантов дизайнерских решений.

### Список использованных источников

1. Сознание и язык. Проблема искусственного интеллекта. [Электронный ресурс]. URL: [https://studopedia.ru/10\\_192176\\_struktura-soznaniya-soznanie-i-samosoznanie.html](https://studopedia.ru/10_192176_struktura-soznaniya-soznanie-i-samosoznanie.html) (дата обращения: 25.05.2020).
2. Эволюционное программирование. [Электронный ресурс]. URL: [http://wi-ki.ru/wiki/Эволюционное\\_программирование](http://wi-ki.ru/wiki/Эволюционное_программирование) (дата обращения: 25.05.2020).
3. Holland, J. H. Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions / J. H. Holland // *Evolutionary Computations*. – 2000 – Vol. 8 (4). – P. 373–391.
4. Sims, K. Artificial Evolution for Computer Graphics / K. Sims, // *SIGGRAPH '91 Proceedings*. – 1991. – С. 319 – 328.
5. Koza, J.R. (1990), Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems, Stanford University Computer Science Department technical report STAN-CS-90-1314.
6. Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / Т. В. Панченко, под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 6 с.
7. Кольчугина, Е. А. Биоинспирированные методы саморазработки программ для распределенных систем / Е. А. Кольчугина // *Известия высших учебных заведений. Поволжский регион. Технические науки*. – 2018 – № 4 (48). – С. 96–104. – DOI 10.21685/2072-3059-2018-4-9.
8. Гладков, Л. А., Курейчик, В. В., Курейчик, В. М. Генетические алгоритмы / под ред. В. М. Курейчика. 2-е изд., испр. и доп. М.: ФИЗМАТЛИТ, 2006. — 320 с.
9. Кольчугина, Е. А., Заваровский, К.В. Применение методов генетического программирования к разработке web-интерфейсов / Е. А. Кольчугина // *Прикладная информатика*. — №5(41), 2012. — Москва, ООО «Синергия Пресс», 2012. — С. 64-74.

10. Саяпин, О. В., Тиханычев, О. В., Чискидов, С. В., Быстракова И. А. Разработка интерфейсов прикладных программ: макетирование или прототипирование / О. В. Саяпин // Прикладная информатика. — №1(85), 2020. — Москва, ООО «Синергия Пресс», 2020. — С. 49–56.
11. Тиханычев О.В. Пользовательские интерфейсы в автоматизированных системах: проблемы разработки // Программные системы и вычислительные методы. – 2019. – № 2. – С. 11 - 22. [Электронный ресурс] URL: [https://nbpublish.com/library\\_read\\_article.php?id=28443](https://nbpublish.com/library_read_article.php?id=28443) (дата обращения: 25.05.2020).
12. Sunghyun, R. K. Webpage design optimization using genetic algorithm driven CSS / R. K. Sunghyun, R. Baer, V. Honavar // Iowa State University. – 2007. – 104 с.
13. Кольчугина Е. А., Заваровский К. В. Разработка методов и средств создания пользовательских интерфейсов с использованием генетического программирования // Современные научные исследования и инновации. Июнь, 2012 [Электронный ресурс] URL: <http://web.snauka.ru/issues/2012/06/15165> (дата обращения: 25.05.2020).
14. Докинз, Р. Эгоистичный ген / Р. Докинз //: Пер. с англ. - М.: Мир, 1993. - 318 с.
15. Пискулин, Д. Е. Применение интерактивного генетического алгоритма для поиска оптимального дизайна web-сайтов на основе CSS / Д. Е. Пискулин. // Молодой ученый. — 2020. — № 11 (301). — С. 24-29. [Электронный ресурс] URL: <https://moluch.ru/archive/301/68054/> (дата обращения: 25.05.2020).
16. ГОСТ Р 54869-2011. Национальный стандарт. Проектный менеджмент. Требования к управлению проектом – Введ. 2012-09-01. – М.: Стандартинформ, 2019. – 9 с.
17. ГОСТ 15.016-2016 СРПП. Межгосударственный стандарт. Техническое задание. Требования к содержанию и оформлению. – Введ. 2017-09-01. – М.: Стандартинформ, 2017. – 53 с.
18. ГОСТ 2.119-73 ЕСКД. Межгосударственный стандарт. Эскизный проект (с Изменениями № 1-5). Общее положение. – Введ. 1974-01-01. – М.: Стандартинформ, 2011. – 11 с.

19. ГОСТ Р ИСО 11442-2014. Национальный стандарт. Техническая документация на продукцию. Управление документацией. – Введ. 2016-01-01. – М.: Стандартиформ, 2018. – 20 с.

20. ГОСТ 25866-83. Государственный стандарт СССР. Эксплуатация техники. Термины и определения (с Изменением №1). – Введ. 1985-01-01 – М.: Издательство стандартов, 1983. – 11 с.

21. ГОСТ Р ИСО/МЭК 14764-2002. Государственный стандарт Российской Федерации. Сопровождение программных средств. – Введ. 2003-07-01 – М.: Издательство стандартов, 2002 – 54 с.

Приложение А  
СХЕМЫ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ  
(рекомендуемое)

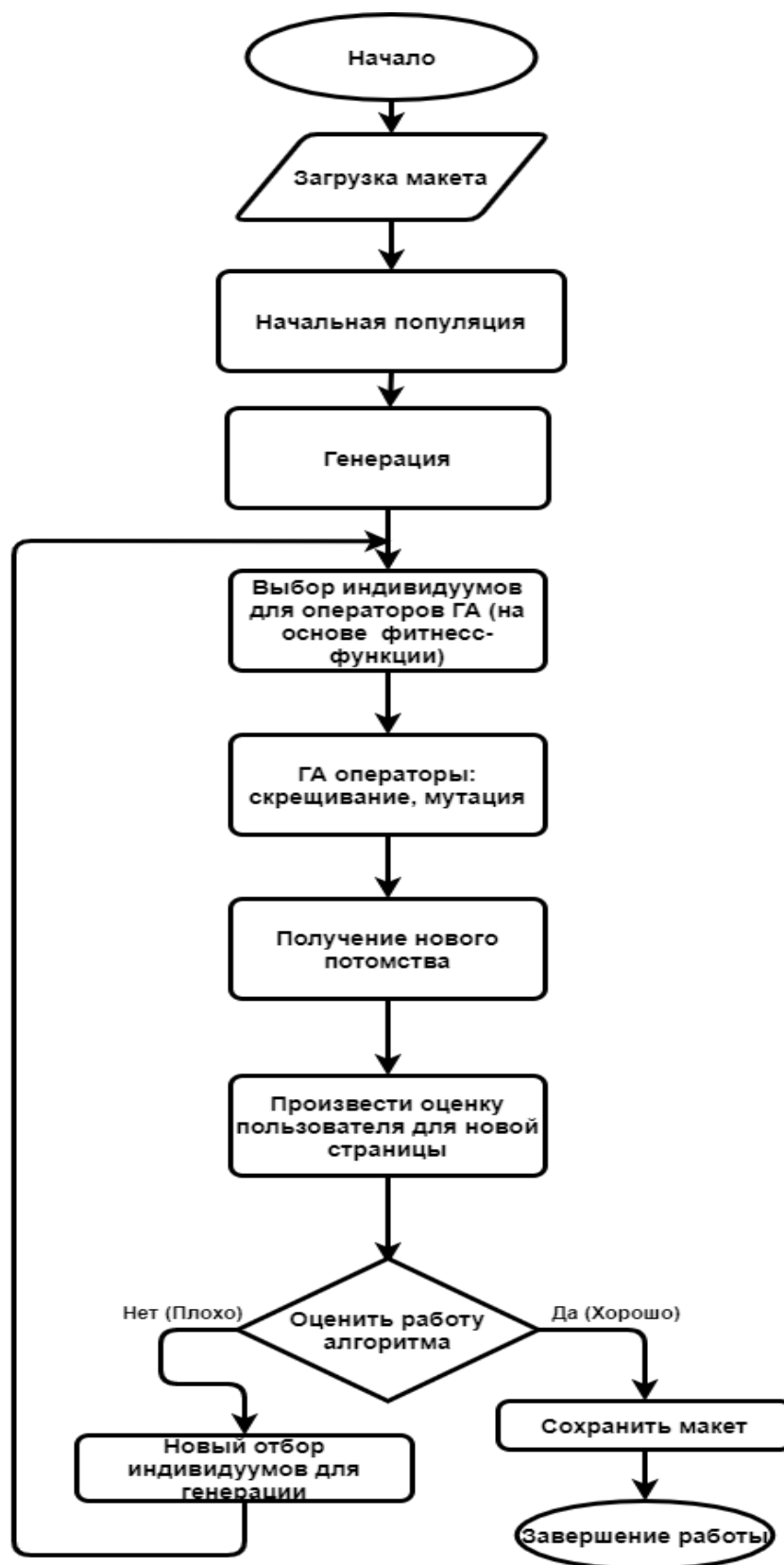


Рисунок А.1 – Блок-схема первого ГА

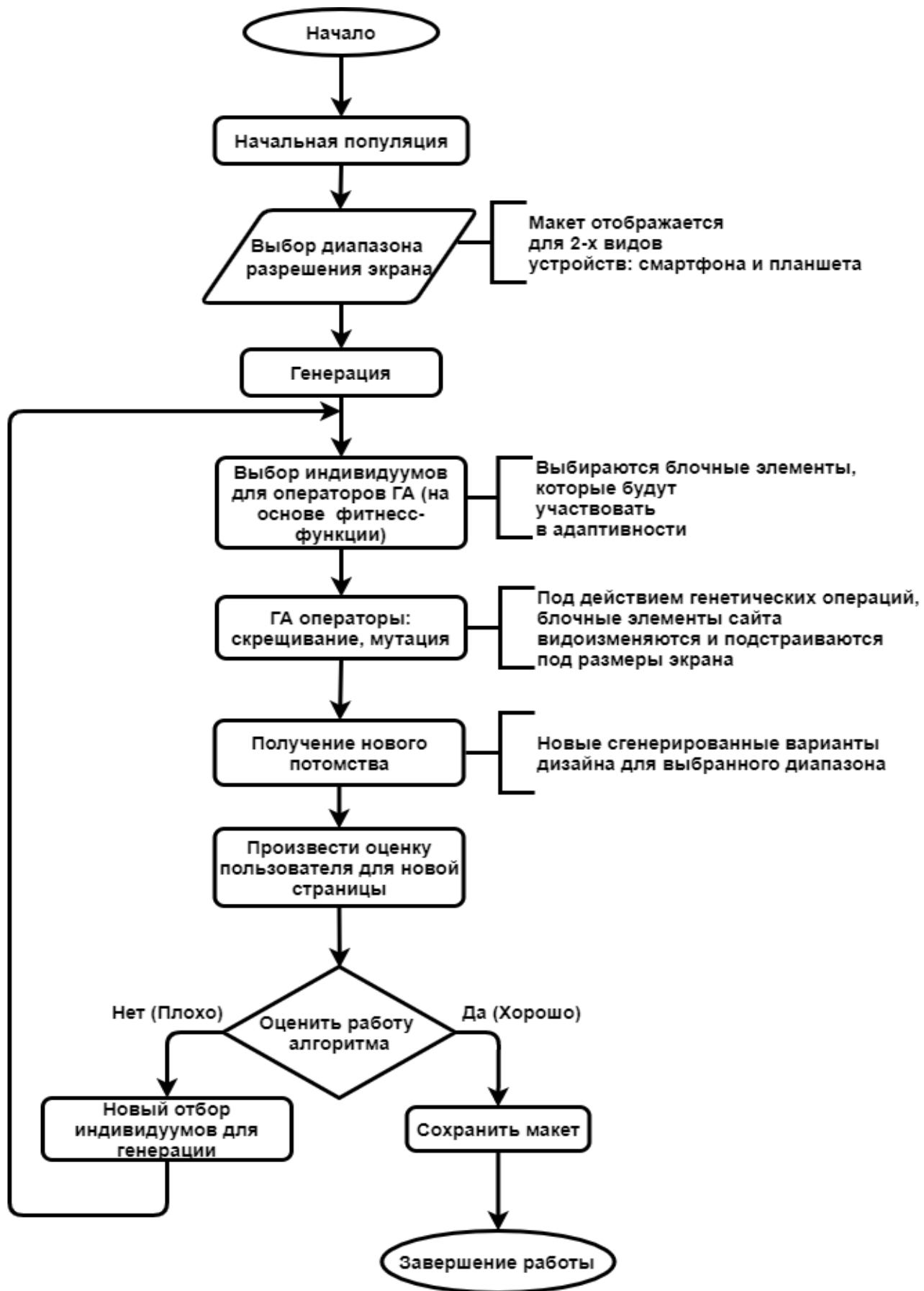


Рисунок А.2 – Блок-схема второго ГА

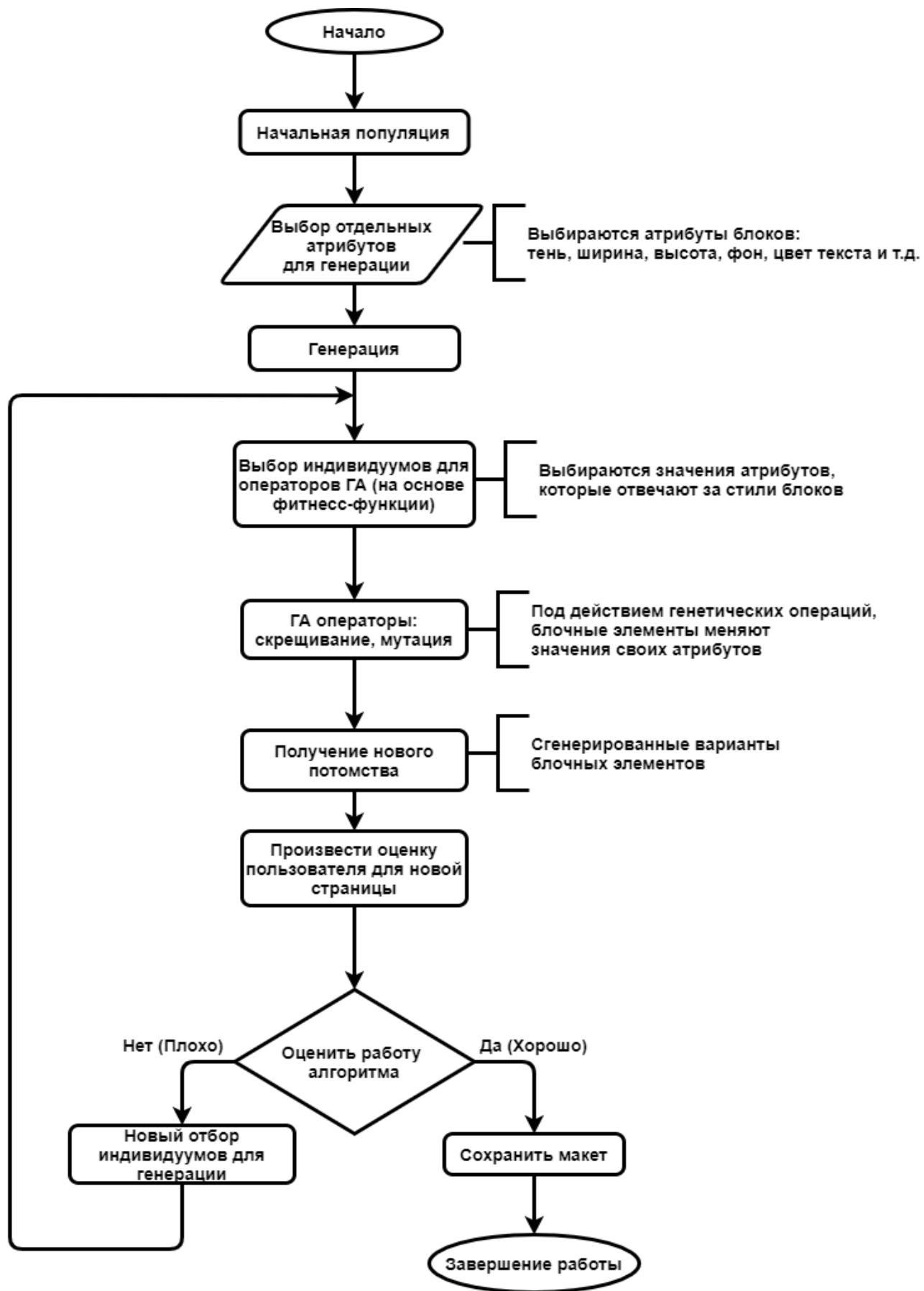


Рисунок А.3 – Блок-схема третьего ГА

Приложение Б

**СГЕНЕРИРОВАННЫЕ ДИЗАЙН-МАКЕТЫ**

(рекомендуемое)

## БЫСТРАЯ ДОСТАВКА

бетона, щебня, песка  
и других нерудных материалов  
по Москве и Московской области

[Подробнее о доставке](#)
[или перейти в каталог](#)

### ВСЕГДА В НАЛИЧИИ



#### Песок речной

Мелкий речной песок повышает прочность, долговечность и морозостойкость растворов, в состав которых он входит.

от 330 р./т

[Подробнее](#)


#### Щебень

Мелкий речной песок повышает прочность, долговечность и морозостойкость растворов, в состав которых он входит.

от 330 р./т

[Подробнее](#)


#### Грунт

Мелкий речной песок повышает прочность, долговечность и морозостойкость растворов, в состав которых он входит.

от 330 р./т

[Подробнее](#)

Рисунок Б.1 – Исходный дизайн-макет

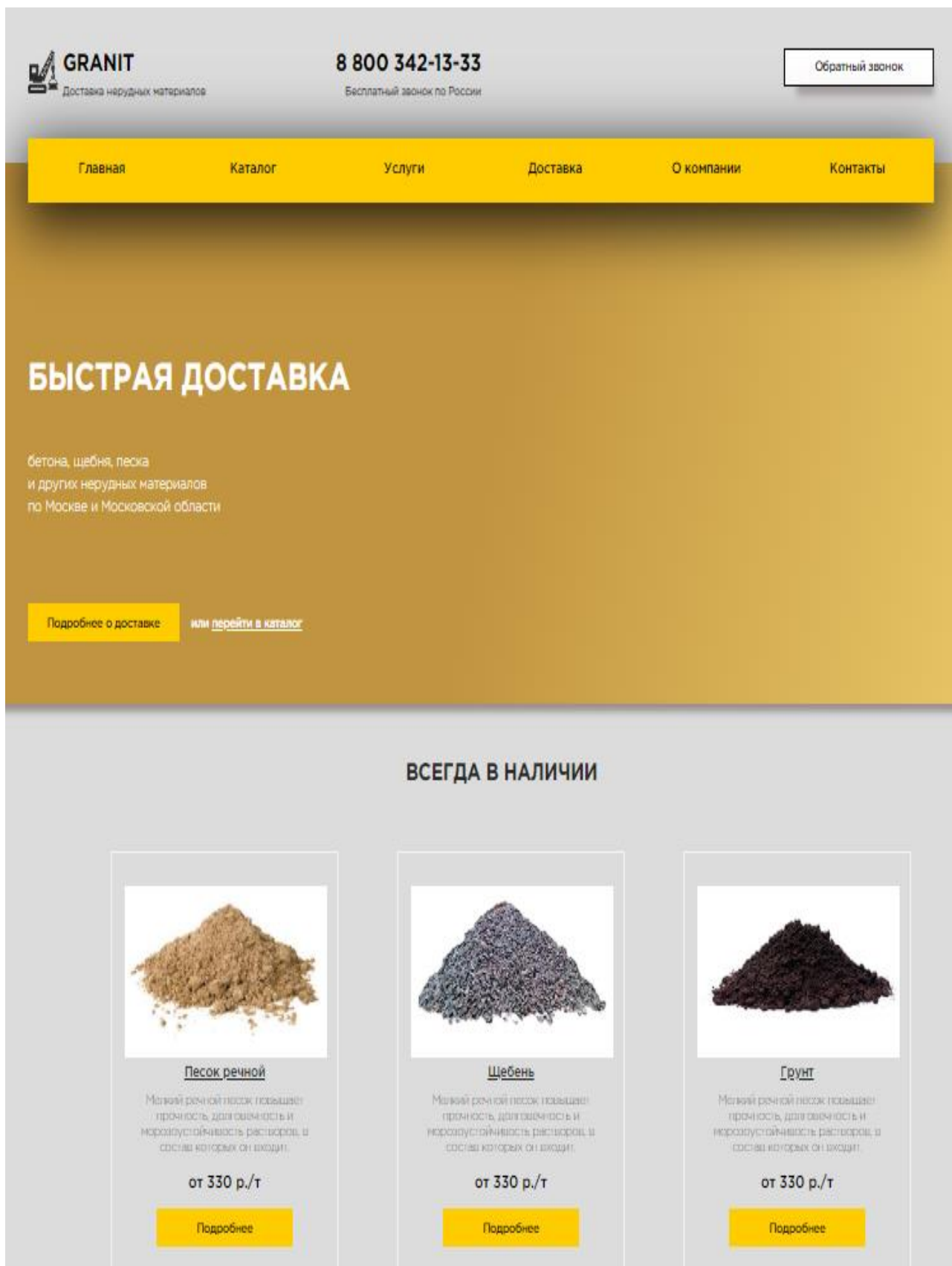


Рисунок Б.2 – Сгенерированный дизайн-макет

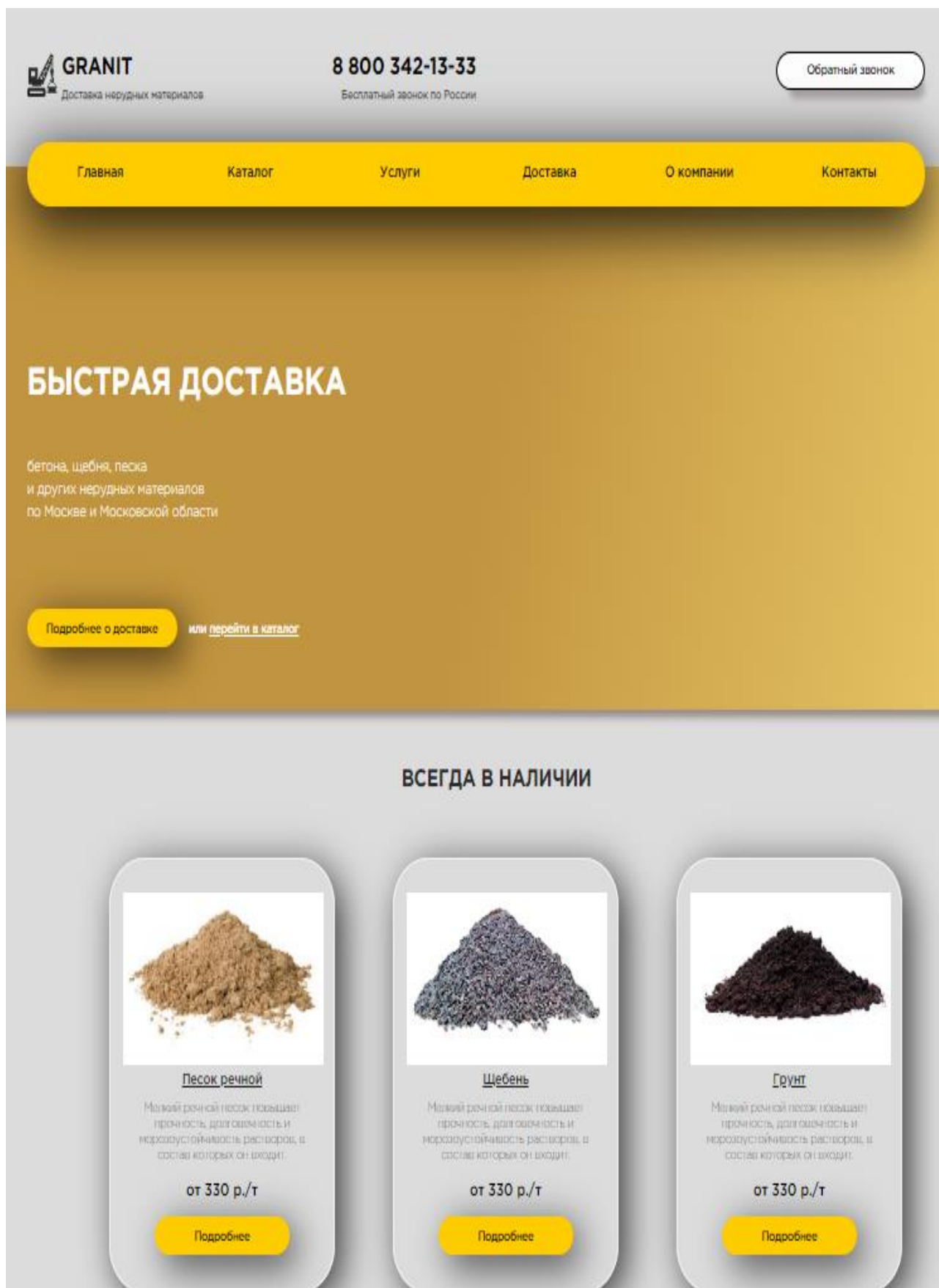


Рисунок Б.3 – Генерация отдельных элементов

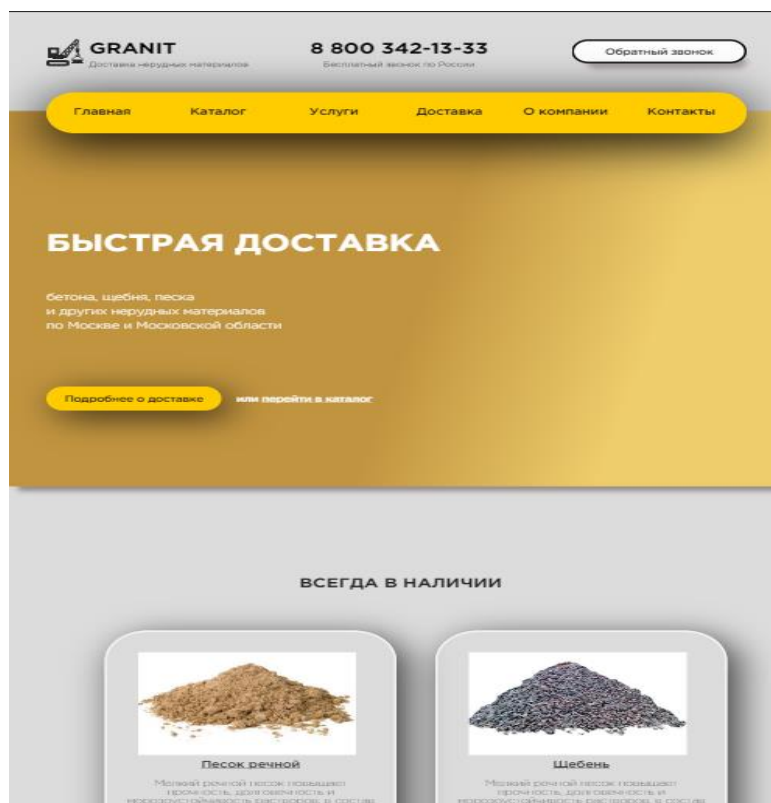


Рисунок Б.4 – Сгенерированный макет под экран планшета



Рисунок Б.5 – Сгенерированный макет под экран смартфона