

**Минобрнауки России**

**Юго-Западный государственный университет**

Кафедра программной инженерии

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ПО ПРОГРАММЕ БАКАЛАВРИАТА**

09.03.04 Программная инженерия

(код, наименование ОПОП ВО: направление подготовки, направленность (профиль))

«Разработка программно-информационных систем»

Разработка web-сайта «Русатом – Аддитивные технологии» на платформе

1С-Битрикс

(название темы)

Дипломный проект

(вид ВКР: дипломная работа или дипломный проект)

Автор ВКР

И. И. Иванов

(подпись, дата)

(инициалы, фамилия)

Группа ПО-926

Руководитель ВКР

А. А. Чаплыгин

(подпись, дата)

(инициалы, фамилия)

Нормоконтроль

А. А. Чаплыгин

(подпись, дата)

(инициалы, фамилия)

ВКР допущена к защите:

Заведующий кафедрой

А. В. Малышев

(подпись, дата)

(инициалы, фамилия)

Курск 2023 г.

# Минобрнауки России

## Юго-Западный государственный университет

Кафедра программной инженерии

УТВЕРЖДАЮ:

Заведующий кафедрой

---

(подпись, инициалы, фамилия)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ г.

### ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ ПО ПРОГРАММЕ БАКАЛАВРИАТА

Студента Иванова И.И., шифр хх-хх-хххх, группа ПО-926

1. Тема «Разработка web-сайта «Русатом – Аддитивные технологии» на платформе 1С-Битрикс» утверждена приказом ректора ЮЗГУ от «07» апреля 2023 г. № 1505-с.

2. Срок предоставления работы к защите «13» июня 2023 г.

3. Исходные данные для создания программной системы:

3.1. Перечень решаемых задач:

- 1) проанализировать IT-инфраструктуру предприятия;
- 2) разработать концептуальную модель системы управления IT-инфраструктурой предприятия на основе подхода к управлению и организации IT-услуг ITSM;
- 3) спроектировать программную систему управления IT-инфраструктурой предприятия;
- 4) сконструировать и протестировать программную систему управления IT-инфраструктурой предприятия.

3.2. Входные данные и требуемые результаты для программы:

- 1) Входными данными для программной системы являются: данные справочников комплектующих, конфигураций, ПО, критериев качества SLA, IT-услуг, департаментов компании; технические данные IT-ресурсов; данные входящих заявок на IT-ресурсы; данные запросов поставщикам на комплектующие.

2) Выходными данными для программной системы являются: сформированные заявки на обслуживание ИТ-ресурсов; сформированные запросы на закупку комплектующих; сведения о выполненных работах по заявкам; статусы заявок; выходные отчеты (инфографика) – по качеству услуг, по состоянию ИТ-ресурсов, по деятельности ИТ-отдела, по стоимости обслуживания ИТ-ресурсов, воронка заявок.

#### 4. Содержание работы (по разделам):

##### 4.1. Введение

##### 4.1. Анализ предметной области

4.2. Техническое задание: основание для разработки, назначение разработки, требования к программной системе, требования к оформлению документации.

4.3. Технический проект: общие сведения о программной системе, проект данных программной системы, проектирование архитектуры программной системы, проектирование пользовательского интерфейса программной системы.

4.4. Рабочий проект: спецификация компонентов и классов программной системы, тестирование программной системы, сборка компонентов программной системы.

##### 4.5. Заключение

##### 4.6. Список использованных источников

#### 5. Перечень графического материала:

Руководитель ВКР

\_\_\_\_\_  
(подпись, дата)

А. А. Чаплыгин

\_\_\_\_\_  
(инициалы, фамилия)

Задание принял к исполнению

\_\_\_\_\_  
(подпись, дата)

И. И. Иванов

\_\_\_\_\_  
(инициалы, фамилия)

## РЕФЕРАТ

Объем работы равен 22 страницам. Работа содержит 4 иллюстрации, 3 таблицы, 0 библиографических источников и 0 листов графического материала. Количество приложений – 2. Графический материал представлен в приложении А. Фрагменты исходного кода представлены в приложении Б.

Перечень ключевых слов: коммерческий сайт, Система, CMS, Битрикс, Joomla, аддитивные технологии, 3D-принтеры, услуги, сервисы, информатизация, автоматизация, информационные технологии, веб-форма, Apache, классы, база данных, средства защиты информации, подсистема, компонент, модуль, сущность, информационный блок, метод, контент-редактор, администратор, пользователь, web-сайт.

Объектом разработки является web-сайт компании, занимающейся производством 3D-принтеров, выпуском оборудования для создания порошков, разработкой программного обеспечения и организацией центров аддитивного производства.

Целью выпускной квалификационной работы является привлечение клиентов, увеличение заказов, информирование о продукции и услугах путем создания сайта компании.

В процессе создания сайта были выделены основные сущности путем создания информационных блоков, использованы классы и методы модулей, обеспечивающие работу с сущностями предметной области, а также корректную работу web-сайта, разработаны разделы, содержащие информацию о компании, ее деятельности, производимой продукции и услугах, разработан сервис по заказу 3D-деталей.

При разработке сайта использовалась система управления контентом «1С-Битрикс: Управление сайтом».

Разработанный сайт был успешно внедрен в компании.

## ABSTRACT

The volume of work is 22 pages. The work contains 4 illustrations, 3 tables, 0 bibliographic sources and 0 sheets of graphic material. The number of applications is 2. The graphic material is presented in annex A. The layout of the site, including the connection of components, is presented in annex B.

List of keywords: commercial website, System, CMS, Bitrix, Joomla, additive technologies, 3D printers, services, informatization, automation, information technology, web form, Apache, classes, database, component, module, entity, information block, method, content editor, administrator, user, web site.

The object of the research is the analysis of information technologies for the development of a production company's website.

The object of the development is the website of a company engaged in the production of 3D printers, the production of equipment for the creation of powders, software development and the organization of additive manufacturing centers.

The purpose of the final qualifying work is to attract customers, increase orders, inform about products and services by creating a company website.

In the process of creating the site, the main entities were identified by creating information blocks, classes and methods of modules were used to ensure work with the entities of the subject area, as well as the correct operation of the website, sections containing information about the company, its activities, products and services were developed, a service for ordering 3D parts was developed.

When developing the site, the content management system «1C – Bitrix: Site Management» was used.

The developed website was successfully implemented in the company.

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ  | 8  |
| 1 Анализ предметной области   | 10 |
| 1.1 Исследование понятия и история появления жанра платформер             | 10 |
| 1.2 Анализ поджанров платформеров   | 11 |
| 1.3 Анализ существующих разработок  | 11 |
| 2 Техническое задание   | 13 |
| 2.1 Основание для разработки  | 13 |
| 2.2 Цель и назначение разработки  | 13 |
| 2.3 Требования пользователя к интерфейсу игры                             | 13 |
| 2.4 Моделирование вариантов использования                                 | 13 |
| 2.5 Требования к оформлению документации                                  | 15 |
| 3 Технический проект  | 16 |
| 3.1 Общая характеристика организации решения задачи                       | 16 |
| 3.2 Обоснование выбора технологии проектирования                          | 16 |
| 3.2.1 Описание используемых технологий и языков программирования          | 16 |
| 3.2.2 Язык программирования Python  | 16 |
| 3.2.2.1 Достоинства языка Python  | 17 |
| 3.2.2.2 Недостатки языка Python   | 18 |
| 3.3 Диаграмма компонентов и схема обмена данными между файлами компонента | 19 |
| 3.4 Диаграмма размещения  | 20 |
| 3.5 Содержание информационных блоков. Основные сущности                   | 21 |

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных.

ИС – информационная система.

ИТ – информационные технологии.

КТС – комплекс технических средств.

ОМТС – отдел материально-технического снабжения.

ПО – программное обеспечение.

РП – рабочий проект.

СУБД – система управления базами данных.

ТЗ – техническое задание.

ТП – технический проект.

UML (Unified Modelling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

## ВВЕДЕНИЕ

Аддитивные технологии (АТ) начали активно развиваться со времени получения первых трехмерных изображений изделий на дисплеях компьютеров. Начала положила стереолитография, затем довольно многочисленные новые принципы стали называть технологиями быстрого прототипирования, затем укоренилось название «Аддитивные технологии». Интенсивность развития данных технологий не имеет аналогов. АТ изменили процессы проектирования и конструирования изделий, превратив их в процессы непрерывного создания изделий. Современные проектирование и производство изделий невозможно представить без данного рода технологий. 3D-принтеры стали такими же распространенными, как и персональные компьютеры. С помощью 3D-принтеров получают ткани, обувь, продукты питания, а также выращивают человеческие органы. Во многих отраслях, например, в космической отрасли, альтернативы аддитивным технологиям нет.

АТ предполагают изготовление детали методом послойного нанесения материала, в отличие от традиционных методов формирования детали, за счёт удаления материала из массива заготовки.

При использовании АТ все стадии реализации проекта от идеи до материализации находятся в единой технологической цепи, в которой каждая технологическая операция выполняется в цифровой CAD/CAM/CAE-системе.

Современные компании, видя, как развиваются информационные технологии, пытаются использовать их выгодно для своего бизнеса, поэтому запускают свой web-сайт. С его помощью предприятие может заявить о себе, проинформировать потенциального заказчика об услугах или продуктах, которые предоставляет, а также позволяет пользователям сделать с помощью сайта онлайн-заказ, произвести покупку или оплатить счета.

Сайт считается лицом компании и может существенно повысить ее имидж. Любой пользователь сети Интернет сможет получить необходимую информацию о компании в любой момент, появляется возможность найти контактные телефоны, адрес и e-mail, чтобы связаться с компанией. Сейчас большинство клиентов узнают



о ее существовании именно через сайт. Поэтому сайт можно назвать самой лучшей рекламой.

Главной задачей профессионально построенного сайта является превращение посетителя, зашедшего на сайт, в потенциального клиента.

*Цель настоящей работы* – разработка web-сайта компании для привлечения новой аудитории, увеличения заказов, рекламы продукции и услуг компании. Для достижения поставленной цели необходимо решить *следующие задачи*:

- провести анализ предметной области;
- разработать концептуальную модель web-сайта;
- спроектировать web-сайт;
- реализовать сайт средствами web-технологий.

*Структура и объем работы.* Отчет состоит из введения, 4 разделов основной части, заключения, списка использованных источников, 2 приложений. Текст выпускной квалификационной работы равен 9 страницам.

*Во введении* сформулирована цель работы, поставлены задачи разработки, описана структура работы, приведено краткое содержание каждого из разделов.

*В первом разделе* на стадии описания технической характеристики предметной области приводится сбор информации о деятельности компании, для которой осуществляется разработка сайта.

*Во втором разделе* на стадии технического задания приводятся требования к разрабатываемому сайту.

*В третьем разделе* на стадии технического проектирования представлены проектные решения для web-сайта.

*В четвертом разделе* приводится список классов и их методов, использованных при разработке сайта, производится тестирование разработанного сайта.

В заключении излагаются основные результаты работы, полученные в ходе разработки.

В приложении А представлен графический материал. В приложении Б представлены фрагменты исходного кода.

## **1 Анализ предметной области**

### **1.1 Исследование понятия и история появления жанра платформер**

Жанр компьютерных игр “платформер” (от англ. platform - платформа) зародился в начале 80-х годов прошлого века и стал одним из первых видов видеоигр. Он был создан в результате эволюции аркадных игр, таких как “Space Invaders” и “Pac-Man”, где игрок управлял персонажем на двухмерной плоскости. Платформеры были популярны на игровых автоматах, а затем перешли на домашние компьютеры и игровые консоли.

Платформер - это жанр, в котором игрок управляет персонажем, перемещающимся по платформам и преодолевающим различные препятствия. Главная задача игрока - пройти уровень от начала до конца, собирая различные предметы и избегая опасностей.

Одним из первых представителей жанра считается игра “Donkey Kong” для аркадного автомата Nintendo, выпущенная в 1981 году. В этой игре игрок управлял человеком, пытающимся спасти свою подружку от обезьяны. Персонаж мог прыгать, а также забираться на платформы и ящики. Игра стала очень популярной, и ее успех привел к созданию многих других платформеров на протяжении следующих десятилетий.

В 1982 году компания Nintendo выпустила игру “Mario Bros.”, в которой игрок управлял братом Марио - Луиджи. Эта игра стала основой для многих будущих платформеров, включая серию “Super Mario Bros.”

Ранние платформеры обычно состояли из уровней, где персонаж должен был преодолеть различные препятствия, такие как пропасти, шипы, враги и ловушки, чтобы достичь конца уровня. Некоторые игры также включали элементы головоломки, где игрок должен был использовать предметы или решать задачи, чтобы продвинуться дальше.

С развитием технологий и графики в середине 90-х жанр платформера стал более разнообразным и детализированным. Появились игры с трехмерной графикой, такие как “Crash Bandicoot” и “Spyro the Dragon”, а также игры с элементами паркура, как “Prince of Persia”.

Сегодня платформеры продолжают оставаться популярными, особенно среди детей и подростков. Они часто включают в себя элементы приключений, головоломок и мультиплеера, а также имеют разнообразные сюжетные линии и персонажей. Многие современные игры сочетают элементы платформера с другими жанрами, такими как гонки, шутеры и ролевые игры, создавая новые и интересные игровые опыты.

Жанр платформера продолжает развиваться и адаптироваться к новым технологиям, оставаясь одним из самых популярных и любимых жанров компьютерных игр.

## **1.2 Анализ поджанров платформеров**

Жанр платформера включает в себя множество поджанров, каждый из которых имеет свои уникальные особенности и характеристики. Вот некоторые из наиболее распространенных поджанров платформеров:

**2D-платформеры:** Это классический поджанр платформеров, который включает в себя игры с двухмерной графикой и управлением. Примеры таких игр включают “Super Mario Bros.”, “Kirby’s Epic Yarn” и “Shantae”.

**3D-платформеры:** Этот поджанр включает в себя платформеры с трехмерной графикой и физикой. Примеры включают “Ratchet and Clank”, “Jak and Daxter” и “Tomb Raider”.

**Метроидвания:** Это платформеры, которые сочетают в себе элементы метроидвании и платформера. Примеры включают “Castlevania: Symphony of the Night”, “Bloodstained: Ritual of the Night” и “Dead Cells”.

**Паркур-платформеры:** Эти игры включают в себя элементы паркура и платформера, такие как прыжки, лазание и бег по стенам. Примеры включают серию игр “Mirror’s Edge”, “Assassin’s Creed” и “Prince of Persia”.

## **1.3 Анализ существующих разработок**

Среди популярных платформеров можно выделить следующие игры:

Super Mario Bros. - классический 2D платформер, в котором игрок управляет персонажем по имени Марио, преодолевая различные препятствия и сражаясь с врагами.

Kirby's Epic Yarn - 2D платформер с уникальной графикой и игровым процессом, где игрок управляет персонажем Кирби, способным менять свою форму и использовать различные способности.

Ratchet and Clank - 3D платформер с элементами экшена и головоломок, в котором игроки управляют двумя героями - Рэтчетом и Кланком, сражающимися с различными врагами и решающими загадки.

Castlevania: Symphony of the Night - игра в жанре метроидвания, сочетающая в себе платформер и исследование мира, где игроки управляют вампиром, способным превращаться в различных существ и использовать разное оружие.

Mirror's Edge - паркур-платформер с видом от первого лица, где игроки могут бегать, прыгать и выполнять различные трюки, преодолевая препятствия и сражаясь с противниками.

Каждая из этих игр имеет свои особенности и уникальный геймплей, который привлекает игроков разных возрастов и предпочтений.

## **2 Техническое задание**

### **2.1 Основание для разработки**

Основанием для разработки является задание на курсовую работу «Разработка компьютерной игры в жанре платформер на языке Python».

### **2.2 Цель и назначение разработки**

Основной целью курсовой работы является разработка компьютерной игры в жанре платформер.

Назначение проекта заключается в развлечении потенциального потребителя.

Задачами данной разработки являются:

- разработка архитектуры приложения;
- разработка интерфейса приложения;
- реализация базовой логики и физики игры;
- создание системы конструктора уровней;
- обработка событий;
- реализация отображения игры на экране.

### **2.3 Требования пользователя к интерфейсу игры**

Игра должна включать в себя:

- возможность управления персонажем;
- графическое оформление интерфейса;
- базовый редактор уровней.

Композиция шаблона сайта представлена на рисунке 2.1.

### **2.4 Моделирование вариантов использования**

Для разрабатываемой игры была реализована модель, которая обеспечивает наглядное представление вариантов использования.

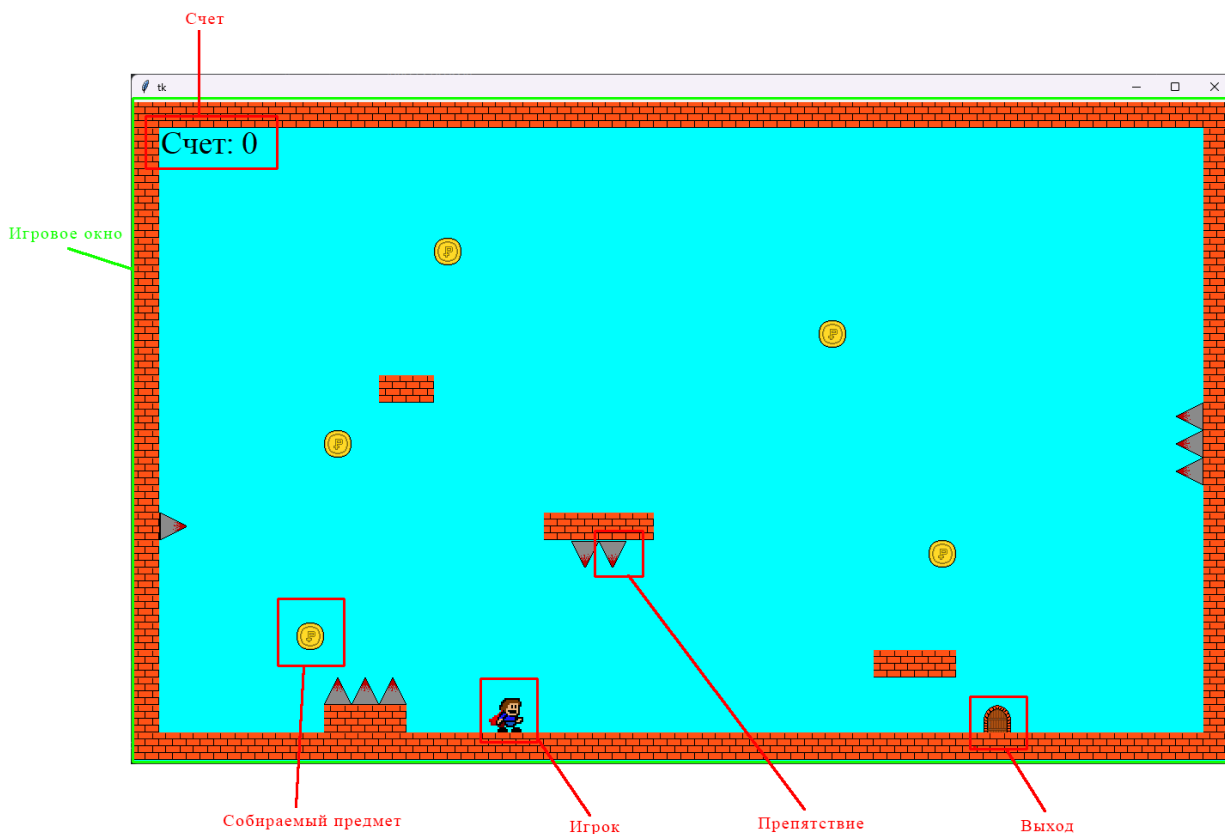


Рисунок 2.1 – Композиция шаблона игры

Она помогает в физической разработке и детальном анализе взаимосвязей объектов. При построении диаграммы вариантов использования применяется унифицированный язык визуального моделирования UML.

Диаграмма вариантов описывает функциональное назначение разрабатываемой системы. То есть это то, что система будет непосредственно делать в процессе своего функционирования. Она является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Проектируемая система представляется в виде ряда прецедентов, предоставляемых системой актерам или сущностям, которые взаимодействуют с системой. Актером или действующим лицом является сущность, взаимодействующая с системой извне (например, человек, техническое устройство). Прецедент служит для описания набора действий, которые система предоставляет актеру.

На основании анализа предметной области в программе должны быть реализованы следующие прецеденты:

1. Запуск игры.
2. Прохождение уровня.
3. Создание уровня.
4. Отображение окна с результатом.

## **2.5 Требования к оформлению документации**

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

### **3 Технический проект**

#### **3.1 Общая характеристика организации решения задачи**

Необходимо спроектировать и разработать игру, которая должна соответствовать всем предъявленным к ней требованиям.

Компьютерная игра — это форма развлечения, предназначенная для запуска и игры на компьютере. Она включает в себя визуальные, звуковые и интерактивные элементы, предоставляя пользователю виртуальное окружение, где тот может взаимодействовать с созданным миром, управлять персонажами и решать различные задачи. Компьютерные игры могут быть разнообразными и включать в себя различные жанры, такие как стратегии, шутеры, головоломки, ролевые игры, платформеры и другие.

#### **3.2 Обоснование выбора технологии проектирования**

На сегодняшний день информационный рынок, поставляющий программные решения в выбранной сфере, предлагает множество продуктов, позволяющих достигнуть поставленной цели – разработки игры.

##### **3.2.1 Описание используемых технологий и языков программирования**

В процессе разработки игры используются язык программирования и программные средства, которые применяются для круга задач, при решении которых они необходимы.

##### **3.2.2 Язык программирования Python**

Питон (Python) - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.



### 3.2.2.1 Достоинства языка Python

Вот некоторые из основных достоинств Python:

#### 1. Простота и читаемость кода:

Python ставит акцент на читаемости кода, что делает его пригодным для быстрого и легкого написания программ. Ясный и понятный синтаксис помогает новичкам и специалистам одинаково.

#### 2. Интерпретируемость:

Интерпретируемость Python обеспечивает быструю разработку и тестирование кода. Без необходимости компиляции, изменения могут быть внесены и протестированы непосредственно.

#### 3. Обширная стандартная библиотека:

Python поставляется с обширной стандартной библиотекой, предоставляя разработчикам множество инструментов и модулей для решения разнообразных задач, таких как работа с файлами, сетевое программирование, обработка данных, создание игр и многое другое.

#### 4. Объектно-ориентированное программирование:

Python поддерживает объектно-ориентированное программирование (ООП), что позволяет создавать модульный и структурированный код, упрощая его повторное использование и сопровождение.

#### 5. Динамическая типизация:

Динамическая типизация Python позволяет гибко использовать переменные, не требуя их явного объявления типа. Это упрощает написание и изменение кода.

#### 6. Поддержка сторонних библиотек и фреймворков:

Существует огромное количество сторонних библиотек и фреймворков для Python, таких как Django для веб-разработки, NumPy и Pandas для анализа данных, TensorFlow для машинного обучения, Pygame для игр, что делает его мощным инструментом в различных областях.

#### 7. Переносимость:

Python является кроссплатформенным, что означает, что программы, написанные на Python, могут быть запущены на различных операционных системах без изменений кода.

### **3.2.2.2 Недостатки языка Python**

Хотя Python является мощным и популярным языком программирования, у него также есть некоторые недостатки, которые могут влиять на выбор его использования в конкретных сценариях. Вот несколько из них:

#### **1. Производительность:**

Python обычно менее эффективен в выполнении определенных задач по сравнению с компилируемыми языками, такими как C++ или Java. Это особенно заметно в вычислительно интенсивных приложениях.

#### **2. Поддержка мобильной разработки:**

Python не является основным языком для мобильной разработки, хотя существуют некоторые фреймворки, такие как Kivy или BeeWare, которые позволяют создавать мобильные приложения на Python. Однако в данном контексте часто предпочитают другие языки, такие как Java (Android) или Swift (iOS).

#### **3. Глобальная блокировка интерпретатора (GIL):**

GIL в Python представляет собой ограничение, которое позволяет только одному потоку исполнения выполняться в определенный момент времени. Это может стать проблемой в многопроцессорных системах, где несколько ядер могут оставаться неиспользуемыми.

#### **4. Объем памяти:**

Python может потреблять больше памяти по сравнению с некоторыми языками программирования из-за своей динамической природы.

**5. Отсутствие некоторых особенностей в сравнении с компилируемыми языками:**

Python может не подходить для разработки низкоуровневых систем или приложений, требующих максимальной производительности и полного контроля над ресурсами

### 3.3 Диаграмма компонентов и схема обмена данными между файлами компонента

Диаграмма компонентов описывает особенности физического представления разрабатываемой системы. Она позволяет определить архитектуру системы, установив зависимости между программными компонентами, в роли которых может выступать как исходный, так и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы, а также зависимости между ними. На рисунке 3.1 изображена диаграмма компонентов для проектируемой системы. Она включает в себя сервер с операционной системой, на которой установлена система управления содержимым, включающая в себя базу данных и интерфейс. Помимо этого на диаграмме изображен клиентский компьютер с операционной системой, на которой установлен браузер.

Любой компонент должен быть вызван в сценарии страницы web-сайта. Web-страница передает данные компоненту в момент вызова последнего.

На рисунке 3.2 представлена схема обмена данными между сценариями компонента при вызове компонента на странице сайта.

При вызове компонента в сценарии web-страницы указываются значения параметров компонента, которые далее посредством массива `$arParams` передаются в сценарий файла `component.php`.

В сценарии файла `component.php` посредством метода `IncludeComponentTemplate` класса `CBitrixComponent` происходит вызов одного из шаблонов компонента. Id шаблона также определяется в сценарии страницы web-приложения и неявно для разработчика передается указанный выше метод. Подключается сценарий файла `template.php` одного из шаблонов, в который передается, возможно, измененный в сценарии `component.php` массив `$arParams` и, также, сформированный в сценарии `component.php` массив `$arResult`. Оба этих массива доступны также и в файле `result_modifier.php`, который подключается перед подключением файла `template.php`.

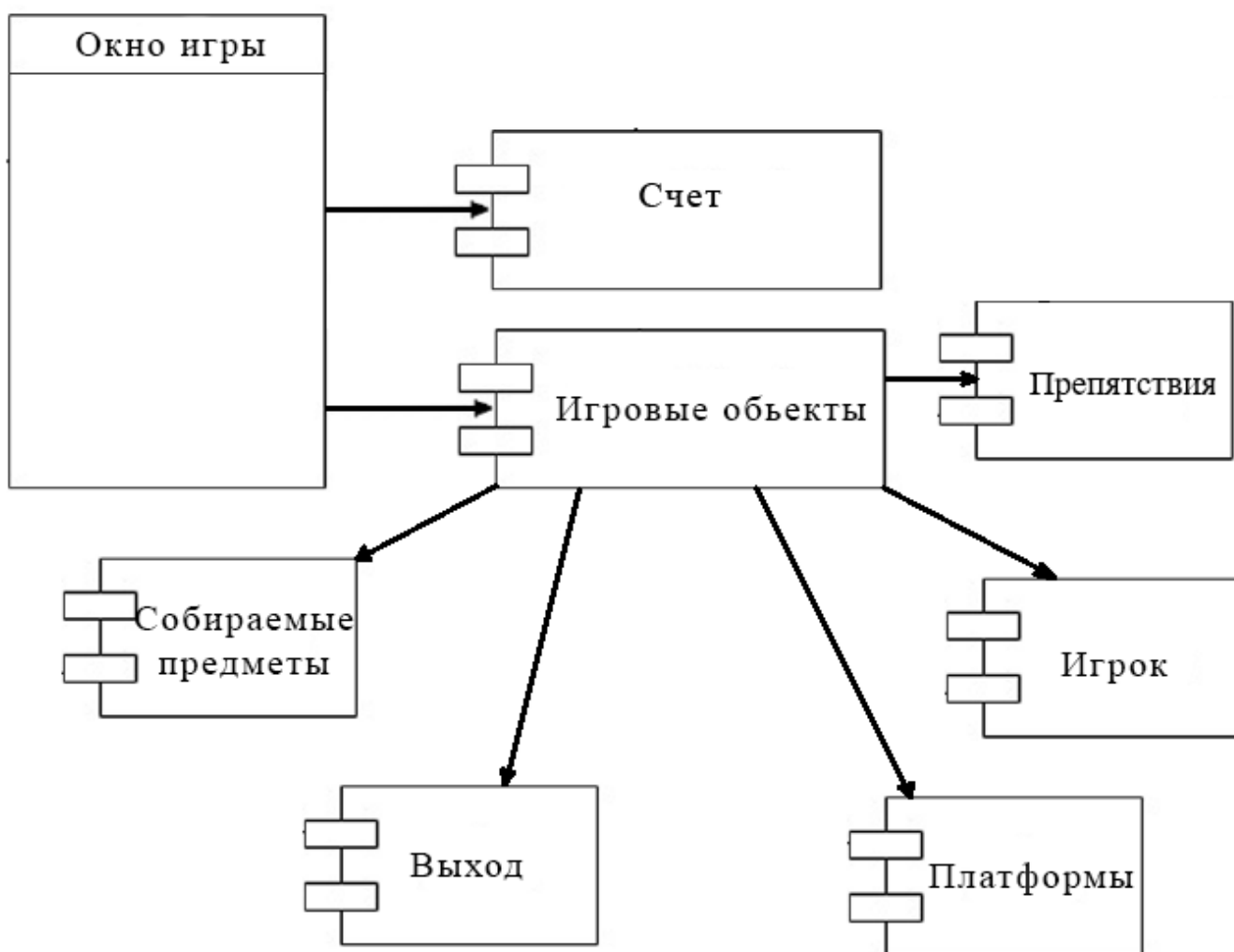


Рисунок 3.1 – Диаграмма компонентов

Работа компонента заканчивается в момент завершения работы сценария файла `component.php`, т.е. возможно выполнить действия уже после подключения шаблона. Однако, если массив `$arResult` будет изменен в сценарии шаблона, в сценарий файла компонента `component.php` измененные данные переданы не будут.

### 3.4 Диаграмма размещения

Диаграмма размещения (рис. 3.3) отражает физические взаимосвязи между программными и аппаратными компонентами системы.

Она является хорошим средством для показа маршрутов перемещения объектов и компонентов в распределенной системе.

В таблице 3.1 приведен пример использования пакета `xltabular` с автоматическим расчетом ширины столбца.

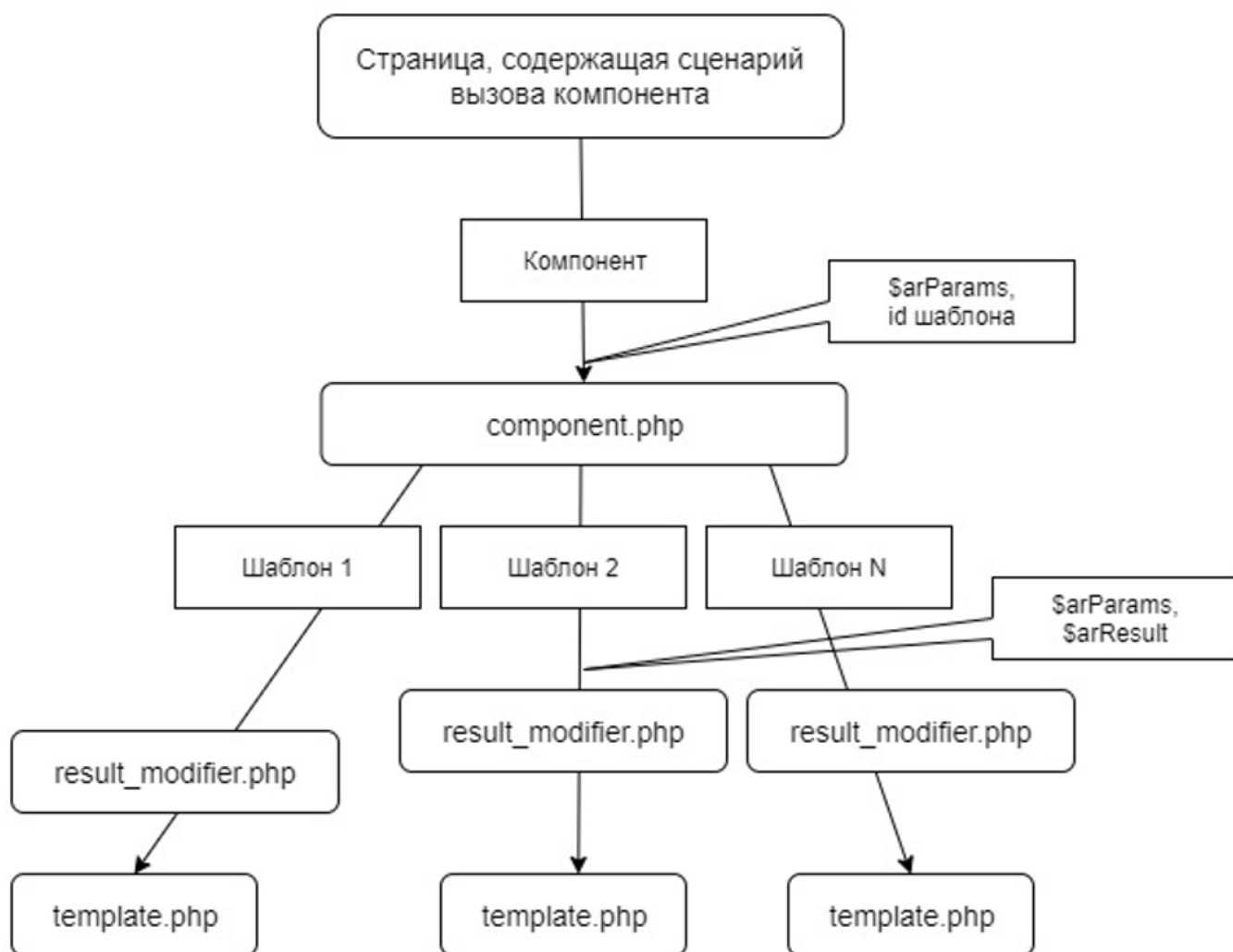


Рисунок 3.2 – Диаграмма компонентов

Таблица 3.1 – Сравнение протоколов SSE и WebSocket

|                  | SSE   | WebSocket   |
|------------------|---|---|
| Направленность   | Однонаправленный, полудуплексный: данные посылает только сервер | Двунаправленный, полнодуплексный: и сервер, и клиент могут обмениваться сообщениями |
| Соединение       | HTTP  | WS  |
| Тип данных       | Только текст  | Бинарные и текстовые данные   |
| Доп. возможности | Встроенный механизм идентификаторов событий и переподключения   | Переподключение и идентификация события реализуются на стороне приложения           |

Продолжение таблицы 3.3

| 1       | 2       | 3     | 4  |
|---------|---------|-------|--|
| author  | String  | false | Автор новости  |
| content | String  | true  | Текст новости  |
| views   | Integer | true  | Количество просмотров новости зарегистрированными пользователями |

Пример использования различных типов столбцов представлен в таблице 3.5. Рекомендуется использовать пакет `xltabular` для создания таблиц.

Таблица 3.5 – Атрибуты сущности «Новости разметки в LaTeX» с использованием различных типов столбцов и многострочным заголовком

| Поле      | Тип      | Обязательное | Описание   |
|-----------|----------|--------------|--|
| 1         | 2        | 3            | 4  |
| _id       | ObjectId | true         | Уникальный идентификатор   |
| head      | String   | true         | Заголовок новости  |
| short     | String   | false        | Аннотация к новости  |
| createdAt | Date     | true         | Время создания новости   |
| author    | String   | false        | Автор новости  |
| content   | String   | true         | Текст новости  |
| views     | Integer  | true         | Количество просмотров новости зарегистрированными пользователями |

В системе предусмотрен внутренний механизм связи между разделами и элементами информационных блоков, поэтому введения дополнительных идентификаторов при реализации связей между сущностями не предполагается.

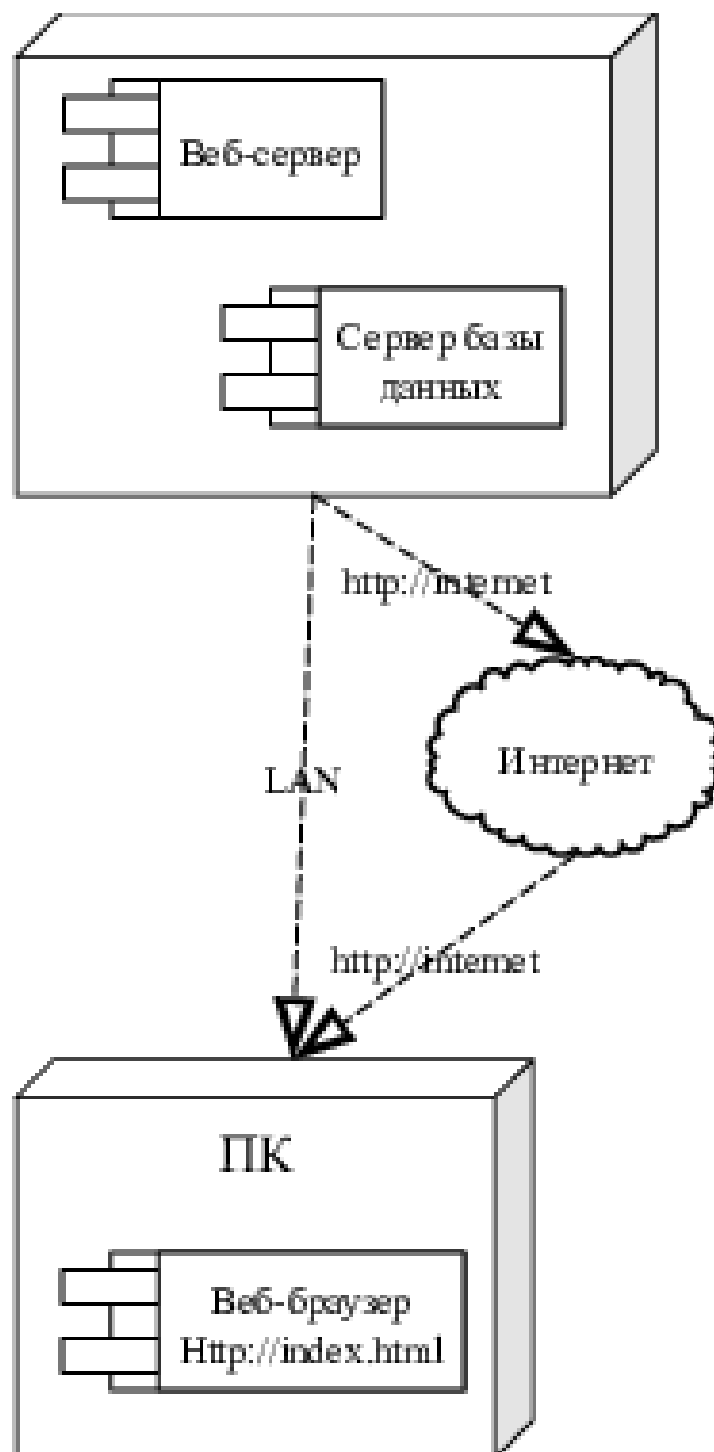


Рисунок 3.3 – Диаграмма размещения. Не помещается на страницу. Очень длинный заголовок

Экземпляры сущностей реализуются в информационных блоках посредством элементов, атрибуты сущности – посредством полей и свойств элемента.