**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное образовательное учреждение города Москвы**

**«Колледж автоматизации и информационных технологий № 20»**

**ДОПУЩЕН К ЗАЩИТЕ**

И.о. заместителя директора

/ И.О. Фамилия

« » 2025 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

**Тема:** **«Разработка информационной системы для автоматизации торговой деятельности индивидуального предпринимателя Ромашова Артема Сергеевича»**

|  |  |
| --- | --- |
| обучающегося группы ИСП411 | Луценко Никиты Романовича |
|  | (Ф.И.О) |

|  |  |  |  |
| --- | --- | --- | --- |
| ⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ |  |  | ⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ |
| (подпись) |  |  | (дата) |

по специальности

09.02.07 Информационные системы и программирование

(код и наименование специальности)

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель проекта | ⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ | А.В. Лисавина | «⠀⠀⠀» ⠀⠀⠀⠀⠀⠀⠀⠀ 2025 г. |
|  | (подпись) | (И.О. Фамилия) |  |
| Консультант по экономической части | ⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ | И.О. Фамилия | «⠀⠀⠀» ⠀⠀⠀⠀⠀⠀⠀⠀ 2025 г. |
|  | (подпись) | (И.О. Фамилия) |  |

**Согласовано**

Заведующий отделением

И.О. Фамилия

« » 2025 г.

2025 г.

**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное образовательное   
учреждение города Москвы**

**«Колледж автоматизации и информационных технологий № 20»**

|  |  |
| --- | --- |
| РАССМОТРЕНО и ОДОБРЕНО  НА ЗАСЕДАНИИ ППО «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ  Протокол №\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Председатель\_\_\_\_\_\_ О.Б. Кузьмина | УТВЕРЖДАЮ  Заведующий отделением  \_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия  «\_\_\_» \_\_\_\_\_\_\_\_ 20\_\_ г. |

**ЗАДАНИЕ**

НА ДИПЛОМНЫЙ ПРОЕКТ ОБУЧАЮЩЕМУСЯ (ЙСЯ)

\_\_\_\_\_\_\_\_\_\_\_Луценко Никите Романовичу\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

|  |  |
| --- | --- |
| 1. Тема: | Разработка информационной системы для автоматизации торговой деятельности индивидуального предпринимателя Ромашова Артема Сергеевича |

1. Тема утверждена приказом по колледжу от 31.01.2025 г. № 15-10/уч.
2. Срок сдачи обучающимся законченного дипломного проекта:«\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2025 г.
3. Исходные данные к дипломному проекту:

Техническое задание на разработку информационной системы.

1. Перечень вопросов, подлежащих разработке в дипломном проекте:

Провести анализ существующих методов оптимизации производственных процессов, разработать архитектуру базы данных и информационную систему, провести тестирование и отладку разработанной системы, разработать руководство пользователя, провести оценку экономической эффективности.

1. Перечень иллюстрированных материалов (графические материалы с точным указанием обязательных чертежей или раздаточных материалов, с указанием наименования таблиц, рисунков или схем):

Диаграмма классов, ER-диаграмма, диаграмма состояний.

Консультанты:

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел работы содержание определяется ППО по направлению** | **Консультант по разделу (ФИО)** | **Дата** | **Подпись** |
| Раздел: Экономическая часть | Иванов Иван Иванович |  |  |

|  |  |
| --- | --- |
| Руководитель дипломного проекта | Задание принял (а) к исполнению обучающийся (щаяся) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  подпись Ф.И.О.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  подпись Ф.И.О.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г. |

**СОДЕРЖАНИЕ**

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 5](#_Toc190519354)

[ВВЕДЕНИЕ 6](#_Toc190519355)

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АВТОМАТИЗАЦИИ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc190519356)

[1.1. Понятие автоматизации и ее основы в торговой деятельности 8](#_Toc190519357)

[1.2. Анализ предметной области 10](#_Toc190519358)

[1.3. Сравнение аналогов информационных систем 12](#_Toc190519359)

[1.4. Определение требований к разработке интерфейса информационной системы 21](#_Toc190519360)

[1.5. Функциональные требования к информационной системе 27](#_Toc190519361)

[1.6. Проблематика внедрения информационной системы 28](#_Toc190519362)

[1.7. Технологический стек разработки 31](#_Toc190519363)

[1.7.1. Язык программирования 32](#_Toc190519364)

[1.7.2. Интегрированная среда разработки 36](#_Toc190519365)

[1.7.3. Система управления базами данных 41](#_Toc190519366)

# СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИИ – искусственный интеллект.

ПО – программное обеспечение

CRM – (от англ. «Customer Relationship Management») – система управления взаимоотношениями с клиентами.

# ВВЕДЕНИЕ

Современный бизнес имеет быстрое развитие рынка и большую конкуренцию, что требует от компаний постоянной адаптации к меняющимся рыночным условиям. Многие предприниматели, по мере увеличения масштабов своего бизнеса, начинают сталкиваться с проблемой улучшения качества обслуживания и дальнейшего расширения клиентской базы. Ключевую роль в этом играет онлайн-торговля. Веб-сайты становятся хорошим инструментом для продвижения своих товаров и услуг, что позволяет оптимизировать свою торговлю и увеличивать доход.

Данный проект имеет большую актуальность в нынешнее время. Огромное количество товаров и услуг приобретается именно в онлайн-режиме, что позволяет многократно увеличить количество клиентов и сделать торговлю эффективнее не только для предпринимателя, но и болей удобной для клиента.

Кроме того, внедрение бизнеса в онлайн индустрию значительно сокращает время, затрачиваемое на повседневные операции, и позволяет сотрудникам выполнять более важные задачи, что приведет к повышению общей эффективности работы компании, а также позволит улучшить качество обслуживания клиентов, повысить качество и количество покупок.

Другие решения в области автоматизации торговой деятельности показывают, что на рынке представлено множество программных продуктов, которые могут удовлетворить потребности бизнеса. Однако многие из них ограничены в функциональности, недостаточно гибки или требуют больших затрат на внедрение и обслуживание. Важно отметить, что каждая организация уникальна и что универсальные решения не всегда вписываются в специфику ее деятельности, поэтому разработка информационной системы представляется наиболее подходящим вариантом.

Ожидается, что разработанная информационная система не только повысит конкурентоспособность предприятия, но и станет важным инструментом устойчивого развития бизнеса в условиях современного рынка.

Объектом исследования является процесс автоматизации торговой деятельности индивидуального предпринимателя.

Предметом исследования является информационная система для оптимизации процесса приобретения товаров и услуг.

Цель – разработка информационной системы для автоматизации торговой деятельности индивидуального предпринимателя Ромашова Артема Сергеевича, обеспечивающей повышение эффективности рабочих процессов и качества обслуживания клиентов.

Исходя из цели были поставлены следующие задачи:

1. Провести анализ существующих методов оптимизации производственных процессов;
2. Разработать архитектуру базы данных и информационную систему;
3. Провести тестирование и отладку разработанной системы;
4. Разработать руководство пользователя;
5. Провести оценку экономической эффективности.

Дипломный проект состоит из введения, …………………..

Объем пояснительной записки составляет 0 страниц, объем списка использованных источников – 0 источников, количество приложений – 0.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АВТОМАТИЗАЦИИ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Понятие автоматизации и ее основы в торговой деятельности

Автоматизация производства – процесс в развитии машинного производство, при котором функции управления и контроля передаются приборам или автоматическим устройствами. Цель автоматизации предприятия – повысить эффективность труда, улучшить качество выпускаемой продукции; создать условия для оптимального использования всех ресурсов производства. Различают следующие виды автоматизации производства: частичную, комплексную и полную.

Частичная автоматизация производства, точнее автоматизирование отдельных производственных операций в тех случаях, когда управление процессами практически недоступно человеку. В других же условиях часть процессов и действий может быть выполнена при помощи простых автоматических устройств или под их заменой другими устройствами. По мере развития средств автоматизации и расширения сферы их применения было установлено, что частичная автоматизация наиболее эффективна тогда, когда производственное оборудование разрабатывается сразу как автоматизированное.

При комплексной автоматизации производства участок, цех, завод, электростанция функционируют как единый взаимосвязанный автоматизированный комплекс. Комплексная автоматизация производства охватывает все основные производственные функции предприятия, хозяйства и службы. Она полезна лишь при высокоразвитом производстве на основе совершенной технологии и прогрессивных методов управления с применением надёжного производственного оборудования по заданной программе; функция человека в этом случае ограничивается общим контролем работы комплекса.

Полная автоматизация производства высшая ступень автоматизации, которая проводится тогда, когда автоматизированное производство рентабельно и устойчиво, а возможные отклонения заранее могут быть учтены при условии отсутствия людей в производственных помещениях или других опасных условиях для жизни человека.

Для нашего предприятия была выбрана частичная система автоматизации в сфере IT технологий, она осуществляется с помощью программных средств и сценариев, которые позволяют автоматически выполнять многие задачи, исключая необходимость ручного вмешательства. Ее применение позволяет предприятиям стандартизировать процессы и повысить согласованность и эффективность работы.

Основные аспекты автоматизации торговой деятельности:

1. **Системы управления торговлей**. Информационные системы позволяют организовать процессы управления запасами, заказами, продажами и аналитикой. Такие системы обеспечивают централизованное хранение данных и автоматизацию ключевых операций. Примеры таких систем включают ERP (Enterprise Resource Planning), которые объединяют несколько бизнес-процессов в единую платформу, обеспечивая прозрачность и контроль на всех этапах;
2. **Автоматизация работы с клиентами. CRM-системы (Customer Relationship Management) помогают отслеживать взаимодействия с клиентами, управлять продажами и формировать персонализированные предложения. Они также предоставляют инструменты для анализа потребностей клиентов, что позволяет повысить эффективность маркетинговых кампаний. Например, такие системы как Salesforce или Bitrix24 включают модули для анализа поведения клиентов;**
3. **Автоматизация бухгалтерского учета и документооборота. Внедрение специализированных решений, таких как «1С: Бухгалтерия», позволяет автоматизировать ведение финансовых операций, формировать налоговые декларации и контролировать денежные потоки. Это особенно важно для соответствия законодательным требованиям и упрощения процессов аудита;**
4. **Мобильные приложения и облачные технологии. Современные решения, например, Google Workspace или Zoho, обеспечивают доступ к данным с любого устройства и в любое время. Это позволяет предпринимателям оперативно реагировать на изменения и принимать решения на основе актуальной информации.**

Автоматизация торговой деятельности индивидуального предпринимателя особенно актуальна, так как позволяет компенсировать ограниченные ресурсы малого бизнеса за счет современных технологий. Кроме того, автоматизация способствует развитию цифровой трансформации бизнеса и повышению конкурентоспособности на рынке.

## Анализ предметной области

Характеристика торговой деятельности индивидуального предпринимателя.

Торговая деятельность индивидуального предпринимателя включает:

1. Закупку товаров у поставщиков;
2. Продажу товаров конечным потребителям через различные каналы (магазины, интернет-платформы, социальные сети);
3. Управление запасами и минимизация затрат на хранение;
4. Ведение бухгалтерской отчетности, налогового учета и документооборота;
5. Анализ продаж для принятия управленческих решений на основе данных.

Особенностью ведения торговой деятельности для индивидуальных предпринимателей является: гибкость и способность быстро адаптироваться к изменениям рынка. Однако ограниченные ресурсы, такие как небольшой штат сотрудников и ограниченный бюджет, делают автоматизацию особенно важной для таких предприятий.

Основные проблемы в ведении торговой деятельности у индивидуальных предпринимателей:

1. **Сложность учета большого объема данных**. Предприниматели сталкиваются с трудностями при отслеживании остатков товаров, учета продаж и анализа покупательского поведения. Это особенно актуально при расширении ассортимента товаров;
2. **Ручное выполнение рутинных операций**. Без автоматизации значительное время уходит на выполнение рутинных задач, таких как ведение отчетности, формирование заказов и учет товарных остатков. Например, процесс инвентаризации может занимать несколько дней;
3. **Ошибки, связанные с человеческим фактором**. Ведение учета вручную увеличивает вероятность ошибок, которые могут привести к финансовым потерям и сбоям в поставках. Например, ошибки в расчетах налогов могут привести к штрафам;
4. **Отсутствие аналитических инструментов**. Без систем автоматизации предпринимателю сложно анализировать ключевые показатели эффективности, такие как объем продаж, средний чек и частота покупок.

Основные задачи автоматизации.

Автоматизация в данном контексте должна решать следующие задачи:

1. Учет и управление запасами на складе с учетом сроков годности и минимального запаса;
2. Ведение базы данных поставщиков и клиентов, включая историю взаимодействий;
3. Отслеживание продаж в режиме реального времени с возможностью аналитики по каналам продаж;
4. Автоматическое формирование отчетов о продажах, доходах и расходах, интеграция с бухгалтерскими системами;
5. Упрощение взаимодействия с клиентами через цифровые каналы, такие как email-рассылки и мессенджеры.

Текущие тенденции в автоматизации торговой деятельности:

1. **Интеграция**[[1]](#footnote-1) **с электронными платежными системами**. Современные системы автоматизации включают поддержку различных методов оплаты, включая QR-коды, бесконтактные платежи и криптовалюты. Это повышает удобство для клиентов и скорость обслуживания;
2. **Использование искусственного интеллекта (ИИ)**. ИИ используется для анализа данных, прогнозирования продаж, определения оптимального уровня запасов и персонализации предложений для клиентов. Например, системы рекомендаций на основе ИИ увеличивают средний чек на 20–30%;
3. **Поддержка омниканальной торговли**. Системы автоматизации позволяют распространять товары по различным каналам, например: физическим точкам, онлайн магазинам, социальным сетям и т.д. При этом информация о клиентах со всех каналов собирается в едином сервисе;
4. Открытие интернет-магазина. Независимость от условий крупных маркетплейсов. Собственный интернет-магазин позволяет избежать высоких комиссий и штрафов маркетплейсов. Полная комиссия крупных универсальных площадок может достигать 40% от стоимости товара, и это без учёта возможных штрафов, которые бывают очень крупными.

## Сравнение аналогов информационных систем

В современном мире происходит активное развитие розничной торговли, и это приводит к усилению конкуренции между торговыми организациями. Это, в свою очередь, заставляет предприятия увеличивать эффективность своей деятельности, что в немалой степени зависит от возможностей используемых программных продуктов по оперативной обработке больших объемов информации. Повышение требований торговых предприятий стимулирует образование новых специализированных программных продуктов с расширенными возможностями. С другой стороны, разработчики программного обеспечения постоянно предлагают торговым организациям все более совершенные продукты, обеспечивая эффективность управления многими бизнес-процессами в торговле. Торговые предприятия часто обращаются к информатизации своей деятельности, при этом во внимание берутся два ключевых фактора: необходимость анализа эффективности деятельности и технологическая необходимость.

Под технологической необходимостью понимаются ситуации, когда внедрение ИТ диктуется требованием обеспечения важных технологических операций. Например, при организации розничной торговли в супермаркете с самообслуживанием задача обработки штриховых кодов обязывает применять информационную систему. Такие же технологические потребности вынуждают оптовые торговые организации внедрять компьютерные программы для автоматизации торговой деятельности. Аналогичная ситуация происходит при анализе эффективности. Потребность в средствах анализа эффективности возникает в основном при наличии большого числа однородных «производственных участков» (филиалы, менеджеры по продажам и т. д.), результаты работы которых можно и нужно постоянно сравнивать.

Современные информационные системы допускают в максимально короткие сроки обнаружить наиболее существенные тенденции развития компании в целом и оценить эффективность работы отдельных структурных подразделений. Сбор, обработка и анализ информации в сроки, максимально приближенные к режиму реального времени, позволяют быстро скорректировать работу компании, а также предусмотреть изменения внешних факторов раньше, чем это сделают конкуренты.

Автоматизация торговой деятельности требует учета специфики выбранной отрасли, включая разнообразие ассортимента, сезонность, управление размерным и цветовым рядом, а также интеграцию с системами маркетинга и аналитики. Рассмотрим основные решения, представленные на рынке.

*«МойСклад»*

Это облачное решение для автоматизации торговли и складского учёта. Преимущественно его используют в малом и среднем бизнесе из-за простоты внедрения, и удобной интеграции с другими сервисами. Сфера применения этой программы очень широка: она подойдет и для розничного магазина, и для парикмахерской, и для кафе или ресторана, и для интернет-площадки. Также можно использовать МойСклад на производстве. Работает с любого компьютера или смартфона, для начала работы не нужны навыки программирования: справитесь, если раньше работали с Excel.

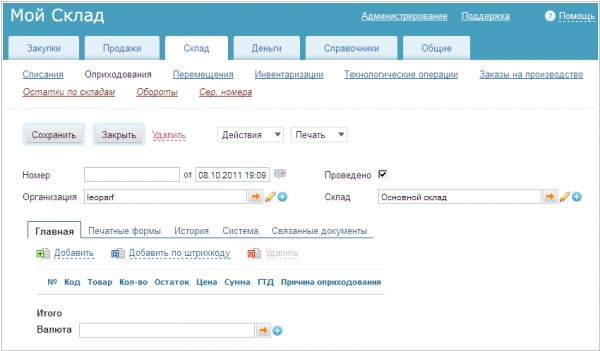


Рисунок 1.1 – Интерфейс системы «МойСклад»

Особенности:

1. Учет цвето-размерных характеристик товаров;
2. Интеграция с кассовыми аппаратами и системами эквайринга;
3. Возможность подключения интернет-магазинов и маркетплейсов;
4. Встроенная аналитика продаж по категориям, брендам, сезонам.

Преимущества:

1. Простота настройки и использования;
2. Доступная стоимость;
3. Подходит для небольших магазинов одежды или индивидуальных предпринимателей.

Недостатки:

Ограниченные возможности для крупных магазинов с широкой филиальной сетью.

*«1С: Управление нашей фирмой»*

Это комплексное решение, предназначенное для бесперебойной автоматизации учета в компаниях, занимающихся мелкосерийным и позаказным производством, оказанием услуг, подрядными работами, оптовой и розничной торговлей, а также складским хозяйством. Система включает в себя все необходимые функции для оптимизации и повышения эффективности деятельности компании. Она предлагает удобный и интуитивно понятный набор инструментов для управления бизнесом и может быть легко настроена в соответствии с конкретными потребностями и требованиями любой компании.

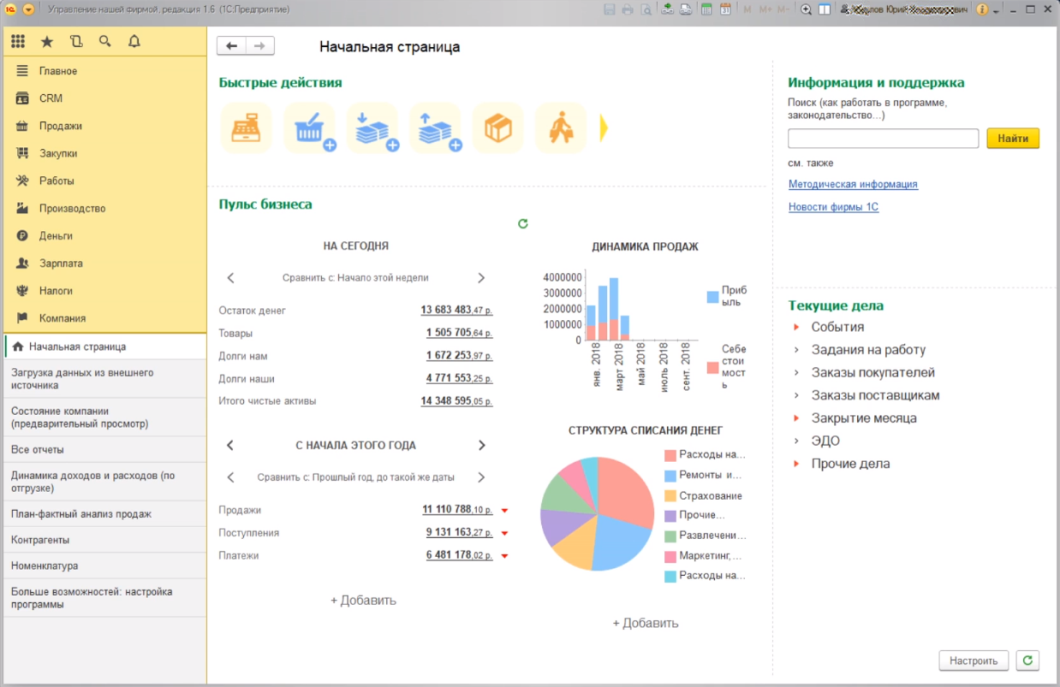


Рисунок 1.2 – Интерфейс системы «1С: Управление нашей фирмой»

Особенности:

1. Поддержка учета по SKU[[2]](#footnote-2) (Stock Keeping Unit), присваивание товарам кода, состоящего из цифр и букв, использующийся для отслеживания остатков на складе, подсчета продаж по группам и категориям товаров и для определения оптимального момента для следующих закупок;
2. Интеграция с программами лояльности и дисконтными системами;
3. Модуль для анализа продаж по сезонам и остаткам на складе.

Преимущества:

1. Широкий функционал, охватывающий практически все аспекты управления магазином;
2. Интеграция с бухгалтерскими продуктами «1С».

Недостатки:

1. Требует определенного уровня подготовки для настройки и работы;
2. Высокая стоимость лицензии и технической поддержки.

*«Эвотор»*

Это широкие возможности базовой функциональности и практически безграничные дополнительные - из магазина приложений. Производитель с партнерами разработал программное обеспечение в соответствии с требованиями законодательства для бизнеса любого направления. Приложения Эвотор расширяют стандартные возможности смарт-терминалов. Есть как платные, так и предоставляемые на безвозмездной основе. Причем у большей части платных продуктов есть пробный период, который позволяет без вложений протестировать [ПО](https://online-kassa.ru/kupit/cat/programmnoe-obespechenie/) и решить, нужно оно или нет.

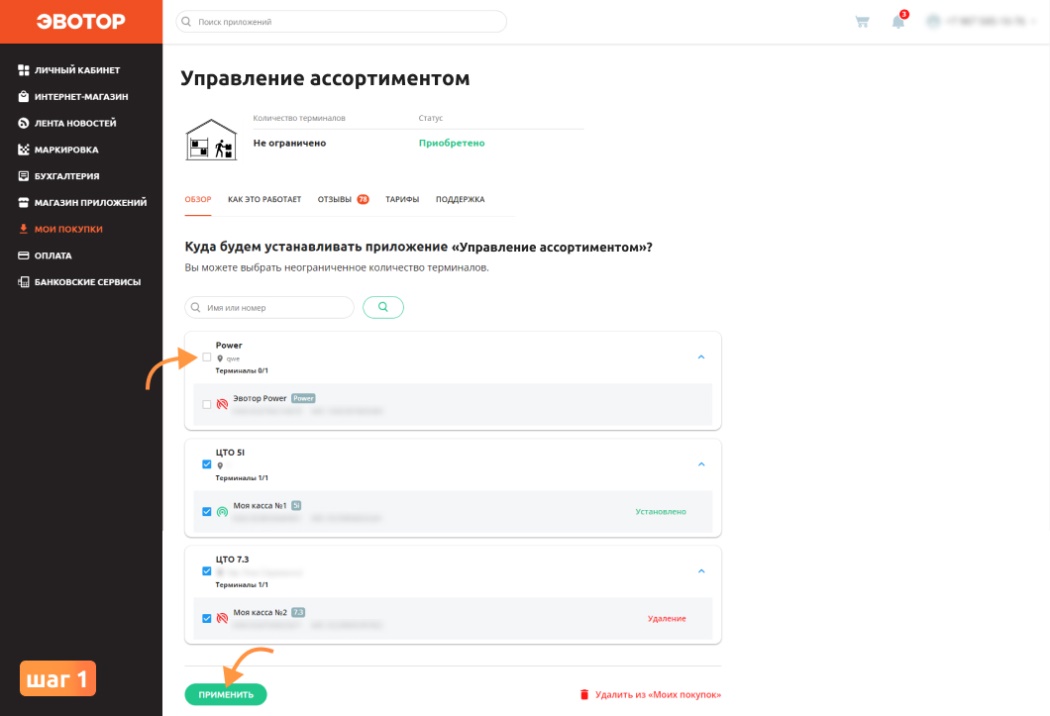


Рисунок 1.3 – Интерфейс системы «Эвотор»

Особенности:

1. Удобные инструменты для работы с кассой и учетом продаж;
2. Возможность настройки учета по цвету, размеру и другим параметрам;
3. Подключение дополнительных приложений для маркетинга и аналитики.

Преимущества:

1. Подходит для малого бизнеса;
2. Простота работы с кассовым оборудованием.

Недостатки:

1. Ограниченные аналитические возможности;
2. Преимущественно используется в розничной торговле без сложных процессов управления складом.

*«Класс365»*

Это облачный сервис для автоматизации бизнеса. Сервис позволяет легко автоматизировать бизнес-процессы розничной и оптовой торговли, сферы услуг, а также организовать эффективную деятельность отдела продаж и склада. Также Класс365 поддерживает интеграцию со многими популярными платформами интернет-магазинов.

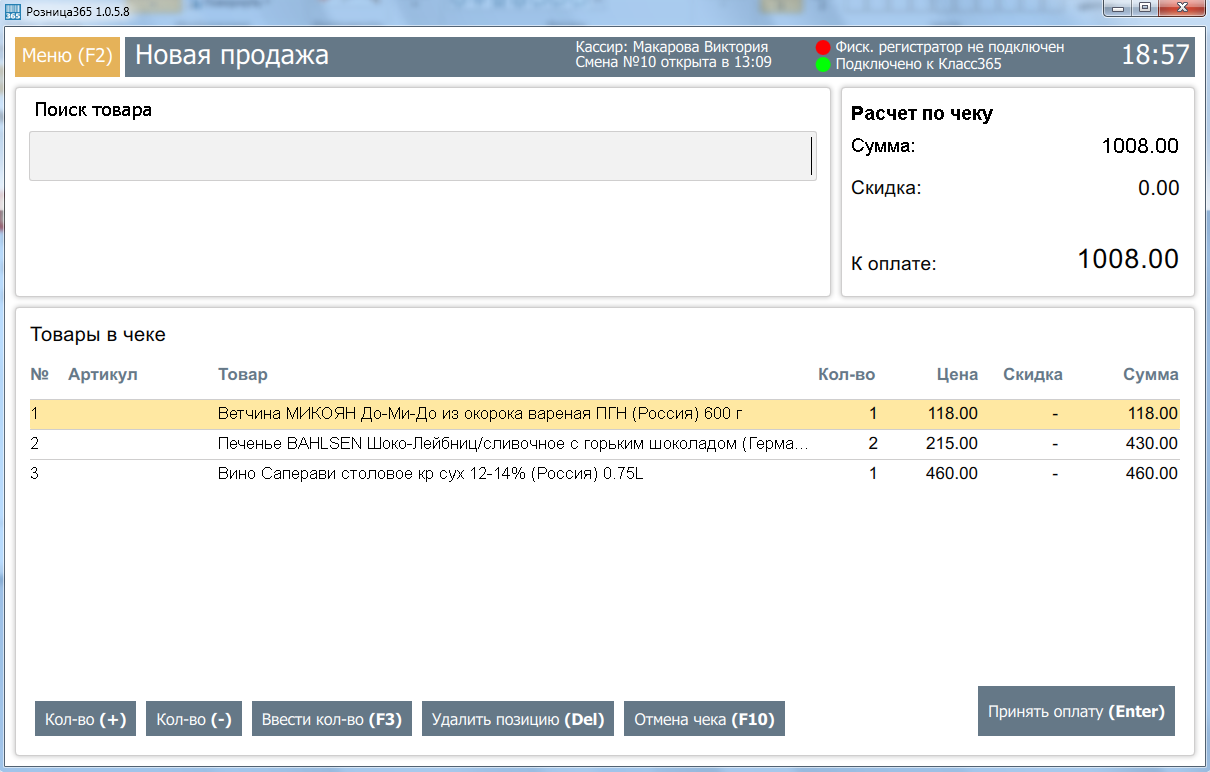


Рисунок 1.4 - Интерфейс системы «Класс365»

Особенности:

1. Полная поддержка учета цвето-размерных характеристик;
2. Возможность подключения к маркетплейсам*[[3]](#footnote-3)*;
3. CRM для работы с клиентами: управление программами лояльности, отслеживание покупательского поведения.

Преимущества:

1. Удобный интерфейс;
2. Облачное решение, доступное с любого устройства.

Недостатки:

Ограниченный функционал для крупных сетевых магазинов.

*«RetailCRM»*

Это решение для eCommerce и ритейла, которое помогает управлять заказами, клиентами и всеми коммуникациями в едином окне. Система автоматизирует и упростит процессы так, что ваши менеджеры смогут обрабатывать больше заказов с лучшей эффективностью.

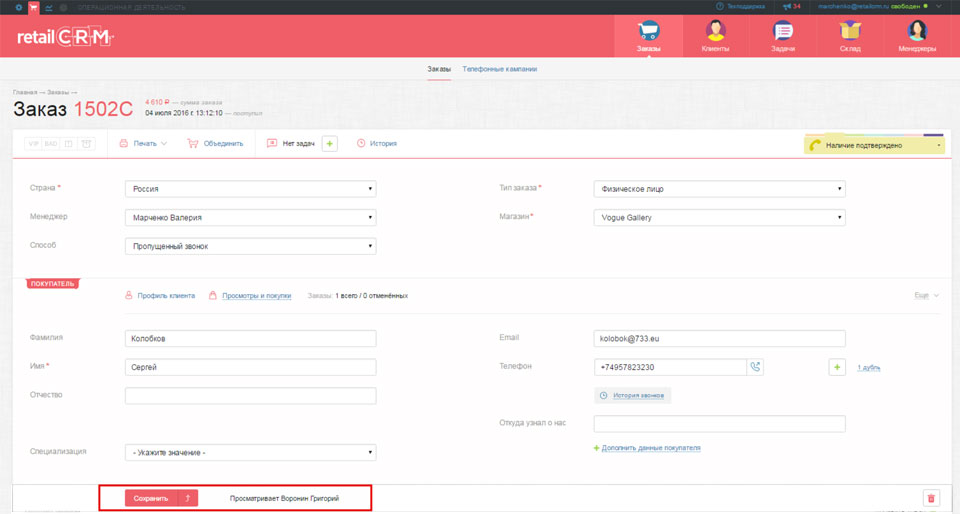


Рисунок 1.5 – Интерфейс системы «RetailCRM»

Особенности:

1. Углубленная CRM-система для работы с клиентами;
2. Поддержка омниканальной торговли: интеграция онлайн- и офлайн-продаж;
3. Аналитика по эффективности акций, кампаний и продаж.

Преимущества:

1. Мощные инструменты для управления клиентскими данными;
2. Возможность персонализированных предложений.

Недостатки:

1. Высокая стоимость;
2. Требует времени на внедрение и обучение.

Таблица 1.1 – Сравнение аналогов информационных систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Система** | **Простота использования** | **Стоимость** | **Функционал для одежды** | **Подходит для малых ИП** | **Подходит для сетей** |
| **МойСклад** | Высокая | Низкая | Средний | Да | Нет |
| **1С: Управление нашей фирмой** | Средняя | Высокая | Высокий | Да | Да |
| **Эвотор** | Высокая | Средняя | Средний | Да | Нет |
| **Класс365** | Высокая | Низкая | Средний | Да | Нет |
| **RetailCRM** | Средняя | Высокая | Высокий | Нет | Да |

Для торговой деятельности, индивидуальных предпринимателей, оптимальными являются такие системы, как «МойСклад», «Эвотор» и «Класс365». Эти решения обеспечивают необходимую функциональность для учета цвето-размерных характеристик, управления складом и продажами, а также обладают доступной стоимостью. Для более крупных магазинов или сетей могут подойти «1С: Управление нашей фирмой» и «RetailCRM», которые предлагают расширенные возможности аналитики и управления клиентскими данными.

*Преимущества собственной системы автоматизации*

Собственная система автоматизации торговли для магазинов одежды обладает рядом преимуществ, которые выгодно отличают её от представленных выше решений:

1. Полная адаптация под конкретные бизнес-процессы:
   1. Учет уникальных требований магазина (специфика ассортимента, сезонность, работа с эксклюзивными брендами);
   2. Возможность добавления нестандартных функций, таких как автоматический расчет скидок по сложным условиям или учет индивидуальных заказов.
2. Гибкость и масштабируемость:
   1. Система разрабатывается с учётом возможного роста бизнеса и расширения ассортимента;
   2. Возможность интеграции с любыми сторонними сервисами (например, редкими маркетплейсами или региональными платёжными системами).
3. Экономия на лицензиях и дополнительных модулях:
   1. Отсутствие необходимости платить за ненужные функции, которые включены в универсальные системы;
   2. Полный контроль над затратами на обслуживание и поддержку.
4. Простота и удобство использования:
   1. Интерфейс разрабатывается с учётом предпочтений и уровня подготовки сотрудников;
   2. Исключение избыточной функциональности для ускорения работы с системой.
5. Безопасность данных:
   1. Локальное или гибридное размещение данных для защиты от утечек;
   2. Реализация индивидуальных механизмов шифрования и резервного копирования.
6. Снижение зависимости от сторонних поставщиков:
   1. Полный контроль над системой без необходимости обращаться к внешним вендорам для внесения изменений.

Собственная система автоматизации торговли, разработанная с учётом конкретных требований индивидуального предпринимателя, позволяет достичь высокой эффективности управления, снизить операционные издержки и обеспечить рост бизнеса. Индивидуальные решения предоставляют уникальные возможности, недоступные в рамках универсальных систем, что делает их оптимальным выбором для предпринимателей, стремящихся к долгосрочной конкурентоспособности.

## Определение требований к разработке интерфейса информационной системы

Интерфейс – это комплекс визуально-программных средств, инструментов, который обеспечивает взаимодействие пользователя с различными электронными устройствами. Этот термин был взят из английского языка (interface) и переводится как взаимодействие, граница, зона контакта. Эту важную составляющую можно сравнить с посредником, который помогает человеку и электронным устройством найти общий язык, прийти к взаимопониманию.

Пользователь при обращении с интерфейсом должен представить себе, какая информация о выполняемой задаче у него существует, и в каком состоянии находятся средства, с помощью которых он будет решать данную задачу. Эффективность работы пользователя и его интерес обеспечивает правильно сформулированная методика разработки и проектирования пользовательского интерфейса.

Именно поэтому необходимо большое внимание уделять процессу построения пользовательских интерфейсов и выстраиванию пользовательского опыта в целом.

Пользовательский интерфейс информационной системы представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными устройствами и аппаратурой.

Иными словами, это тот набор кнопок, ссылок, форм, диалоговых окон, иконок, пиктограмм, баннеров, ползунков и лент прокрутки, с помощью которого пользователь управляет продуктом.

Интерфейс – только половина во взаимодействии с системой, другая половина – человек, пользователь. Для хорошей работы интерфейса нужно точно знать, что именно в любой конкретный момент пользователь воспринимает в интерфейсе, о чем думает, чего хочет добиться.

Интерфейсы являются основой взаимодействия всех современных информационных систем.

Визуально привлекательный и удобный пользовательский интерфейс - ключевой показатель качества сайта. В сочетании с грамотной структурой и логичной навигацией по разделам ресурса, он привлекает посетителей и улучшает функциональность сайта. Главная задача в такой работе, как проектирование интерфейса информационной системы – максимально упростить жизнь пользователю, сделать так, чтобы он достигал желаемый результат, затрачивая минимум усилий.

Рассмотрим преимущества правильно спроектированного пользовательского интерфейса:

1. Повышение конкурентоспособности;
2. Снижение стоимости разработки;
3. Увеличение аудитории продукта;
4. Уменьшение затрат на обучение и поддержку пользователей;
5. Уменьшение потерь продуктивности работников при внедрении системы и более быстрое восстановление утраченной продуктивности;
6. Доступность функциональности системы для максимального количества пользователей;
7. Снижение риска ошибок.

Для достижения этих целей, используется множество особенностей пользовательского интерфейса, самыми главными из них, являются простота и удобство его использования. Рассмотрим эти понятия более подробно.

Удобство и простота интерфейса в информационной системе – одни из самых важных аспектов, влияющих на успешность и эффективность работы конечного пользователя с системой. Правильно организованный интерфейс значительно повышает пользовательский опыт, облегчает выполнение задач и способствует тому, чтобы система стала интуитивно понятной и доступной для самых разных пользователей. Вот почему удобство и простота интерфейса имеют такое важное значение:

*Повышение эффективности и производительности*

Удобный и простой интерфейс способствует тому, что пользователи могут быстро и без усилий выполнять свои задачи. Когда интерфейс интуитивно понятен, пользователю не нужно тратить время на поиски нужных функций или изучение сложных инструкций. Это особенно важно в информационных системах, где эффективность и скорость выполнения операций напрямую влияют на продуктивность работы. Например, если интерфейс позволяет пользователю выполнить задачу в два клика вместо пяти, это сэкономит время и уменьшит вероятность ошибок.

*Снижение нагрузки на пользователя*

Интерфейс, лишённый избыточных или ненужных элементов, позволяет избежать перегрузки пользователя информацией. Когда система проста в использовании, внимание пользователя не отвлекается на второстепенные элементы, а сосредоточено на ключевых действиях. Это снижает когнитивную нагрузку, позволяя пользователю быстрее адаптироваться и сфокусироваться на выполнении конкретных задач, не чувствуя себя перегруженным.

*Повышение удовлетворенности пользователей*

Положительный пользовательский опыт - результат того, что интерфейс понятен и логичен, что значительно повышает удовлетворённость пользователя системой. Удовлетворённый пользователь скорее вернется к системе, будет использовать её с регулярностью, а также рекомендовать её другим. Важно помнить, что, если пользователи сталкиваются с трудностями или непониманием в процессе работы, они могут потерять интерес к продукту или даже отказаться от его использования. Простота и удобство интерфейса служат гарантией того, что система будет восприниматься как надежная и эффективная.

*Снижение числа ошибок и упрощение обучения*

Удобный интерфейс помогает избежать ошибок, которые могут возникнуть из-за путаницы или непонимания работы системы. Логичная структура, очевидные и понятные элементы управления, ясные указания на то, что и как нужно делать, минимизируют вероятность пользовательских ошибок. К тому же, когда интерфейс прост и понятен, пользователю легче обучиться работе с системой, независимо от его уровня технической подготовки. Это сокращает время на обучение и повышает уверенность в использовании системы.

*Доступность для широкого круга пользователей*

Важным аспектом удобства является то, что интерфейс должен быть понятен и доступен для пользователей с разными уровнями подготовки. Не все пользователи имеют одинаковый опыт работы с технологиями, и важно, чтобы система была доступна для новичков, а также для опытных специалистов. Простота и интуитивность интерфейса позволяют минимизировать барьер входа для пользователей, не обладающих техническими знаниями, и сделать использование системы доступным для широкого круга пользователей.

*Улучшение взаимодействия между пользователями и системой*

Простой и удобный интерфейс позволяет пользователю сосредоточиться на содержании и функциональности системы, а не на том, как её использовать. Это улучшает взаимодействие с системой, делает его более плавным и безошибочным. Чётко продуманная логика интерфейса способствует тому, чтобы взаимодействие с системой приносило положительный опыт.

Кроме удобства и простоты интерфейса для пользователя, так же не стоит забывать о красоте, элегантности и гармоничности его восприятия.

Программисты заботятся о функциональности приложения, однако внешний вид его может влиять на подсознательное поведение пользователей при работе с приложением. Поэтому в приложениях, которые помогают пользователям выполнить серьезную задачу, можно поставить акцент на задаче, сохраняя декоративные элементы тонкими и ненавязчивыми, и управлять ими с помощью стандартных средств управления предсказуемого поведения. Эти приложения посылают четкие, унифицированные сообщения о своей цели и своей идентичности, которая помогает людям использовать их.

Если приложение, решая серьезные задачи, выглядит навязчивым и легкомысленным, то пользователи могут подвергнуть сомнению его надежность. С другой стороны, в приложениях, таких как игра, пользователи ожидают увидеть красочный внешний вид, который обещает получение удовольствия и радости от игры. И это вызывает желание открыть такое приложение. Люди не ожидают серьезного внешнего вида от игры. Внешний вид игры и поведение должны соответствовать ее назначению.

Рассмотрим какие принципы могут быть использованы в дизайне интерфейса. Для того чтобы интерфейс был эстетически привлекательным, необходимо, чтобы он был незаметен в процессе его использования. Для удобства использования и красоты нужно, чтобы интерфейс был неощущаемый. Для этого:

1. Стоит избегать излишней красочности и резкости. Лучше, чтобы интерфейс был скромнее;
2. Также не нужно использовать ярких цветов. Существует очень немного цветов, обладающих и яркостью, и мягкостью (т. е. «не бьющих по глазам»). На экране их значительно меньше, поскольку в жизни такие цвета обычно моделируются как собственно цветом, так и текстурой, с чем на экране есть проблемы;
3. Избегать острых углов в изображении;
4. Стараться сделать изображение максимально более легким и воздушным для восприятия пользователя;
5. Добиваться контраста не сменой насыщенности элементов, а расположением пустот.

Разработка визуальной части информационной системы является важнейшим этапом, который в значительной степени определяет успех всего продукта. Это не просто создание «красивой картинки», а решение комплексных задач по удобству, доступности и функциональности. Интерфейс системы должен быть не только визуально привлекательным, но и удобным, интуитивно понятным и доступным для различных категорий пользователей. Он должен обеспечивать быстрый и комфортный доступ к необходимым данным и функционалу, создавать положительный пользовательский опыт и снижать вероятность ошибок.

Создание качественного визуального интерфейса – это всегда результат слаженной работы дизайнеров, разработчиков, тестировщиков и специалистов по пользовательскому опыту. Применение лучших практик в разработке интерфейсов и учёт потребностей пользователей помогут создать систему, которая будет не только высокофункциональной, но и успешной на рынке. В конечном счете, успешная визуальная часть информационной системы может стать тем фактором, который определит её популярность, лояльность пользователей и успех в долгосрочной перспективе.

## Функциональные требования к информационной системе

В современном мире индивидуальные предприниматели нуждаются в эффективных инструментах для автоматизации процессов управления бизнесом. Основная цель разработки информационной системы – обеспечение удобного и функционального решения для выполнения таких задач, как ведение учета клиентов, заказов, финансов, а также анализа данных. Данная информационная система будет реализована в виде веб-приложения[[4]](#footnote-4) с архитектурой клиент-сервер.

Клиентская часть отвечает за взаимодействие пользователя с интерфейсом и должна быть интуитивно удобной и понятной, а серверная – за обработку данных, выполнение бизнес-логики и взаимодействие с базой данных.

Для пользователя с ролью «Клиент» должны быть предусмотрены следующие функциональные требования:

1. Возможность авторизации новых пользователей;
2. Просмотр информационной системы без авторизации;
3. Добавление товаров в корзину;
4. Оформление заказа для авторизированного пользователя;
5. Просмотр списка товаров;
6. Поиск по списку товаров.
7. Контакты для обращения в случае возникновения вопросов у пользователей;

Данная информационная система с клиент-серверной архитектурой будет полезным инструментом в развитии малого бизнеса. Удобный интерфейс и широкая функциональность обеспечат высокую эффективность и комфорт в использовании системы.

## Проблематика внедрения информационной системы

В современном мире ни один средний или крупный бизнес не обходится без информационных технологий или автоматизированных информационных систем. Эффективность автоматизации бизнеса невозможна без внедрения информационной системы. Её наличие – это показатель благополучия компании, согласованности всех модулей управленческой и производственной цепочек, начиная от руководителя и заканчивая простыми рабочими. Основные задачи информационной системы понятны – автоматизация бизнеспроцессов компании и функций управления. Автоматизированная информационная система также позволяет в значительной степени снизить финансовые и временные издержки компании. Внедрение информационной системы на предприятии является довольно сложным и зачастую болезненным процессом, как и любое другое новшество. Однако некоторые сложности, которые могут возникнуть при внедрении новой системы, достаточно хорошо изучены, и на данный момент существуют даже эффективные методики разрешения этих проблем. Основным препятствием для внедрения автоматизированной информационной системы является то, что большинство сотрудников не готово к обучению работе с новой системой, и поэтому сопротивляется нововведениям.

Также при планировании внедрения автоматизированной информационной системы возникает ряд вопросов, которые связанны с техническим, экономическим и организационным аспектом. Технические вопросы внедрения автоматизированной информационной системы связаны с необходимостью создания или совершенствования инфраструктуры, для того чтобы обеспечить интеграцию новой системы с уже внедренными технологическими решениями. К экономическим проблемам, возникающим при внедрении автоматизированной информационной системы можно отнести то, что в новую систему требуется вложить много средств, которые могут окупиться не сразу. Организационные вопросы связаны с отсутствием мотивации у работников компании, так как это мешает приступить к работе с новой системой, требует повышения уровня технических знаний для грамотного анализа потребностей клиентов.

Заблаговременное изучение проблем, которые могут возникнуть при внедрении системы и подготовка к ним значительно могут облегчить процесс и повысить эффективность дальнейшего использования автоматизированной информационной системы.

Основными проблемами и задачами, возникающими в большинстве случаев при внедрении информационной системы управления компанией, требующих особого внимания при их решении являются:

1. Участие топ-менеджера в разработке плана внедрения автоматизированной информационной системы;
2. Отсутствие постановки задачи менеджмента в компании;
3. Необходимость в частичной или полной реорганизации структуры компании;
4. Необходимость изменения технологий бизнеса в различных аспектах;
5. Сопротивление сотрудников предприятия;
6. Временное увеличение нагрузки на сотрудников при внедрении информационной системы;
7. Формирование квалифицированной команды по внедрению и сопровождению системы, а также выбор грамотного руководителя рабочей команды.

Перед тем, как приступить к внедрению автоматизированной информационной системы в компании, требуется частичная реорганизация его структуры и технологий ведения бизнеса. Для этого особо важным этапом проекта внедрения системы будет полный анализ предприятия во всех элементах его деятельности. На основе полученной информации и строится в дальнейшем схема автоматизированной информационной системы. Безусловно, можно автоматизировать уже существующую структуру организации, но этого не следует делать по ряду причин. В результате анализа зачастую выявляются места возникновения необоснованных дополнительных затрат, а также несоответствия в организационной структуре. Устранение выявленных проблем позволило бы уменьшить производственные и логистические издержки, а также значительно сократить время реализации различных этапов основных процессов.

Также на определенных этапах реализации проекта, нагрузка на работников компании возрастает. Связанно это с тем, что помимо основных обязанностей, сотрудникам необходимо осваивать новые технологии и совершенствовать свои знания.

Формирование квалифицированной команды по внедрению и сопровождению системы, а также выбор грамотного руководителя рабочей команды.

Внедрение большинства крупных автоматизированных информационных систем реализуется по определенной технологии. В компании формируют рабочую группу, которая проходит полное обучение работе с внедряемой системой, затем перед этой группой ставится задача по внедрению системы и дальнейшему ее сопровождению.

Данная технология применяется по двум причинам:

* + - 1. Компания обычно заинтересована в том, чтобы у нее в штате сотрудников были специалисты, которые смогут решать возникшие вопросы при настройке и эксплуатации системы;
      2. Обучение своих сотрудников значительно дешевле, чем привлечение внешних источников.

Формирование высококвалифицированной рабочей группы является залогом успешной реализации проекта по внедрению информационной системы.

Исходя из вышесказанного, можно сделать вывод о том, что в современном мире бизнеса руководители проявляют значительный интерес к автоматизации управления деятельностью предприятия; важно привлекать специалистов, которые смогут эффективно решать проблемы, связанные с внедрением или использованием автоматизированной информационной системы, по мере их поступления. Для грамотного использования информационной системы и увеличения рентабельности предприятия требуется повышение квалификации сотрудников предприятия. Несмотря на все возникающие проблемы при внедрении, информационная система направлена на повышение качества работы предприятия. Следовательно, соблюдение рекомендаций по уже существующим методологиям поможет снизить риски убытков компании.

## Технологический стек разработки

Выбор технологического стека – это один из ключевых этапов в разработке информационной системы, который определяет не только её функциональность и производительность, но и успешность всего проекта в целом. От него зависит, насколько легко система будет разрабатываться, поддерживаться, масштабироваться и удовлетворять запросам пользователя.

### Язык программирования

При выборе языка программирования для разработки информационной системы необходимо учитывать следующие критерии:

1. **Производительность и скорость работы:** важно, чтобы язык обеспечивал высокую скорость обработки данных и выполнения вычислений;
2. **Простота разработки и читаемость кода:** чем легче освоить язык, тем быстрее можно приступить к разработке. Хорошая структура и синтаксис языка упрощают чтение и поддержку кода;
3. **Поддержка многопоточности и асинхронности:** необходима возможность эффективной обработки многозадачности (например, запросы к серверу, параллельные вычисления);
4. **Экосистема и инструменты разработки**: развитая экосистема ускоряет разработку. Хорошие среды разработки упрощают процесс программирования;
5. **Возможности интеграции:** лёгкость подключения к базам данных, API и сторонним сервисам.

*C#*

Язык программирования С# (C-sharp) – является на данный момент самым популярным и современным языком программирования компании Майкрософт.

Он обладает расширенной функциональностью, а также увеличивает скорость разработки программного обеспечения по сравнению со своими конкурентами.

С# работает со всеми известными СУБД (MS SQL Server, MySql, Oracl и т.д.). К тому же, компания Microsoft распространяет его бесплатно в пакете Visual Studio.

Из этого следует, что его можно скачать с официального сайта компании и использовать в научных целях.

Из плюсов можно выделить:

1. Высокая производительность благодаря компиляции в промежуточный код и оптимизированному выполнению в .NET Runtime;
2. Отличная поддержка многопоточности и асинхронности;
3. Поддержка кроссплатформенности с выходом .NET Core;
4. Хорошая интеграция с Microsoft SQL Server, PostgreSQL, REST API;
5. Современные механизмы безопасности.

Однако данный инструмент разработки программного обеспечения имеет свои недостатки:

1. C# очень требователен к характеристикам компьютера, в особенности к оперативной памяти;
2. Скорость работы программы, написанной на языке C#, является низкой по сравнению с конкурентами в этой области;
3. Отсутствует гибкость в разработке (сложно установить устойчивое взаимодействие с неизвестными инструментами), а также не подходит для многих сфер программирования, например, таких как программирование микроконтроллеров.

*Java*

Язык программирования Java – наряду с языком C# является одним из самых популярных языков в мире. Также, как и С#, является бесплатным для решения научных и внутриорганизационных задач, но в отличии от C# является кроссплатформенным языком программирования, т.е. он может работать на операционных системах Windows, Linux, Mac OS и т.д. Работает со всеми широко применяемыми СУБД.

К плюсам данного языка можно отнести:

* 1. Кроссплатформенность: работает везде, где есть JVM;
  2. Высокая производительность;
  3. Хорошая поддержка многопоточности;
  4. Богатая экосистема (Spring, Hibernate, JavaFX).

Но Java также имеет существенные недостатки:

Язык Java, как и С#, разработан американской компанией, что не укладывается в рамки политики импортозамещения;

1. Java очень требователен к характеристикам компьютера, из-за использования своей виртуальной машины;
2. Отсутствует гибкость в разработки;
3. Низкая скорость работы программы.

*Python*

Python − высокоуровневый интерпретируемый язык программирования с простым синтаксисом. Применяется в веб-разработке, анализе данных, машинном обучении и автоматизации. Простой и легкий в освоении синтаксис Python обеспечивает удобство использования.

Python поддерживает модули и пакеты, что поощряет модульность программы и повторное использование кода. Язык Python и обширная стандартная библиотека доступны бесплатно для всех основных платформ и могут распространяться свободно.

К плюсам Python относятся:

1. Простота и удобство для изучения. Python − это язык со строгой логикой с очень простым и понятным синтаксисом, что делает его очень простым в работе для начинающих программистов;
2. Широкие возможности и библиотеки. Python имеет много различных библиотек и фреймворков[[5]](#footnote-5) для работы в различных областях, включая науку о данных, машинное обучение, веб-разработку и другие;
3. Платформенная независимость. Python может быть запущен на большинстве платформ, включая Windows, MacOS, UNIX, Linux.

Недостатками данного языка являются:

1. Низкая производительность из-за интерпретации кода;
2. Высокая нагрузка на процессор при работе с многопоточной обработкой данных;
3. Ограниченные возможности для высоконагруженных многопоточных серверов.

*C/C++*

Язык С/С++ может работать со всеми известными СУБД. Компиляторы этого языка распространяются бесплатно, что существенно упрощает вопросы лицензирования, а единый стандарт делает его полностью кроссплатформенными. Преимуществом С/С++ является то, что он может напрямую вызывать машинный язык Assembler, что позволяет ускорить проводимые вычисления, а также снизить аппаратные требования к компьютеру.

К плюсам можно отнести:

1. Высокая производительность и контроль над памятью;
2. Широкие возможности в системном программировании и играх;
3. Используется в разработке критически важных систем.

Но и С++ не лишен недостатков, вот самые распространенные из них:

Высокий порог вхождения. Это самая большая проблема С/С++, программист должен обладать высокой квалификацией, чтобы эффективно использовать все возможности данного инструмента;

1. Время на разработку программного обеспечения уходит больше, чем у его конкурентов, таких как Java или C#, это связано с его универсальностью. Из представленного множества языков программирования, наиболее оптимальным для решение научных задач является С/С++, так как из-за прямого взаимодействия с препроцессором обладает одной самых высоких скоростей выполнения программного кода. В скорости он уступает только языку Assembler. Это позволяет на нем писать инженерные и математические программы.

Таблица 1.2 – Сравнительная характеристика языков программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **С#** | **Java** | **Python** | **C++** |
| **Производительность** | + | + | + | + |
| **Простота разработки** | - | + | + | – |
| **Многопоточность** | + | + | + | + |
| **Экосистема** | + | + | + | – |
| **Интеграция** | + | + | + | – |

При выборе языка программирования, было принято решение остановиться на языке Python, такой выбор был сделан из-за личных предпочтений, опыта работы, наибольшей практикой в освоении, простоты использования и возможности интеграции с другими технологиями.

### Интегрированная среда разработки

Так же следует определиться с наиболее подходящей средой разработки для поставленных задач. Поскольку языком программирования был выбран Python, выбранная среда разработки должна поддерживать использование данного языка и удобство его использования.

При выборе среды разработки были выставлены следующие критерии:

**Производительность и быстродействие:** важно, чтобы среда разработки быстро запускалась и не замедляла процесс написания кода. Также важна оптимизация работы с большими проектами и сложными структурами;

1. **Поддержка отладки и инструментов анализа кода**: наличие встроенного отладчика и пошагового выполнения кода. Поддержка анализа кода для выявления ошибок и потенциальных проблем;
2. **Поддержка расширений и экосистемы для удобной работы с фреймворками, HTML и СSS.**
3. **Кроссплатформенность: возможность использования полной версии IDE и всех ее функций на разных операционных системах.**
4. **Удобство интерфейса:** гибкая настройка среды, поддержка горячих клавиш, наличие встроенного терминала, автодополнения кода, умных подсказок.

*Visual Studio*

Данная среда разработки дает возможность разрабатывать как консольные приложения, веб-приложения, так и ПО с поддержкой графического интерфейса для ПК и мобильных устройств. Подходит для специалистов, работающих с несколькими языками программирования. VS является мультиязычной средой, поддерживающей Python, PHP, JavaScript, JSON, HTML, C++, и другие. Тестировщики же могут проверять приложение на всех возможных устройствах и браузерах. С помощью дополнительных модулей есть возможность расширить возможности среды под конкретные задачи.

Преимущества:

1. Постоянно расширяющийся магазин дополнений, предоставляющий надстройки для расширения возможностей взаимодействия с IDE[[6]](#footnote-6);
2. Встроенная технология автозавершения кода IntelliSense;
3. Возможности для командной работы: управление доступами к проекту, полностью настраиваемые параметры редактора;
4. Мультиязычность;
5. Индивидуальная настройка рабочей панели;
6. Есть поддержка split screen.

Недостатки:

1. Отсутствие возможности работы на Linux ОС;
2. Это достаточно тяжеловесная среда. Если у вас нет необходимости разрабатывать консольные приложения, то можно рассмотреть более подходящий редактор кода.
3. Не является полноценной кроссплатформенной средой, предназначена в основном для Windows, для других операционных систем имеется урезанная в функционале версия.

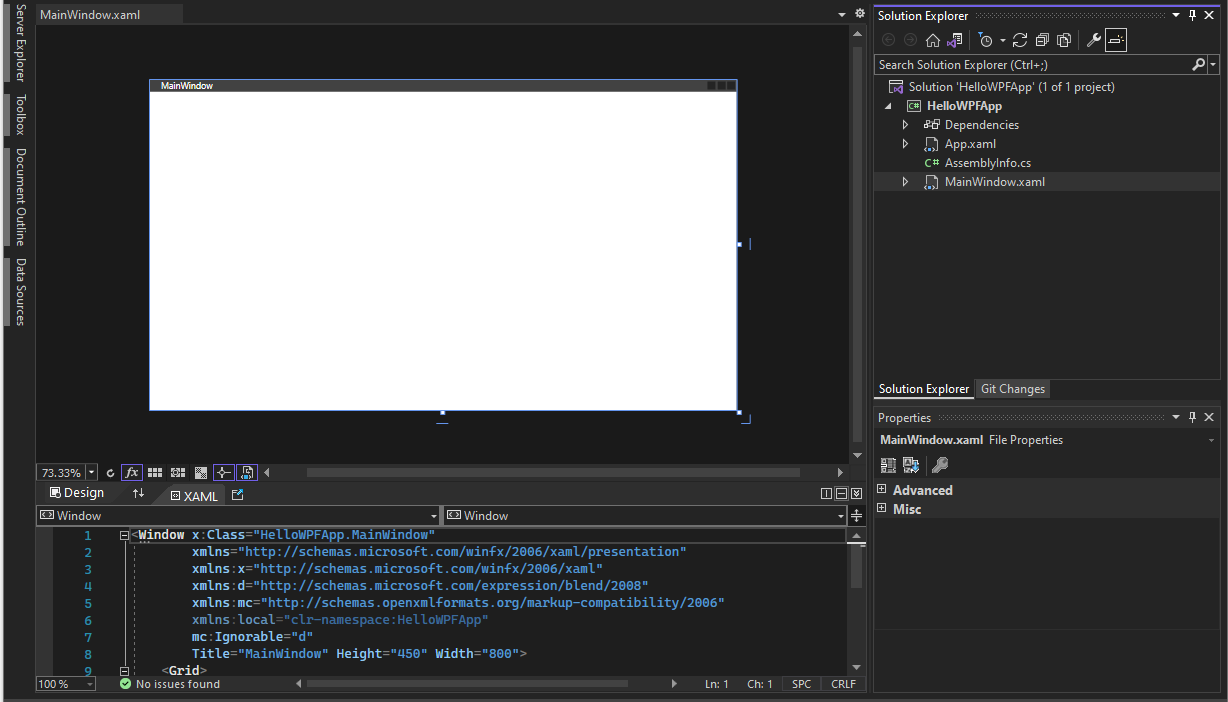


Рисунок 1.6 – IDE Visual Studio

*Visual Studio Code*

Это бесплатный редактор кода, который включает в себя инструменты запуска и отладки, интеграцию с системой контроля версий и дает возможность писать код на многих языках программирования: С#, C++, CSS, JavaScript, JSON, PHP, Python и другие.

Преимущества:

1. Кроссплатформенность: поддержка на Windows, macOS и Linux;
2. Встроенная технология автодополнения кода IntelliSense;
3. Множество бесплатных расширений в [маркетплейсе](https://marketplace.visualstudio.com/VSCode" \t "_blank);
4. Полностью бесплатный редактор.

Недостатки:

1. В Visual Studio Code доступно меньше инструментов, чем в полноценной IDE от Microsoft;
2. Отсутствует поддержка split screen.

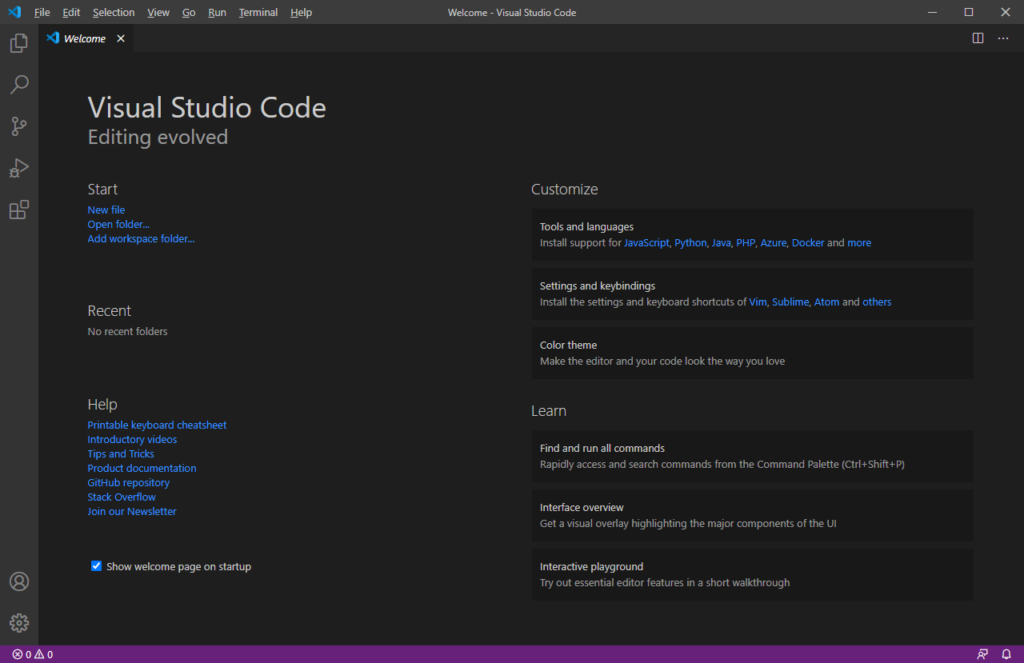


Рисунок 1.7 − IDE Visual Studio Code

*PyCharm*

Это мощная кроссплатформенная IDE для Python, разработанная компанией JetBrains. PyCharm предлагает широкие возможности для разработки веб-приложений. IDE обладает высокой производительностью, интегрированным отладчиком, а также поддерживает использование других языков программирования.

Преимущества:

1. Особенно силен в работе с Python. Он предоставляет мощную встроенную поддержку шаблонов, моделей и маршрутов, что ускоряет разработку и уменьшает количество ошибок. IDE отлично справляется с автодополнением кода, предлагает быстрые исправления и подсвечивает потенциальные ошибки ещё до запуска программы. Это делает работу более безопасной и продуктивной.
2. Встроенный отладчик в PyCharm — один из самых удобных среди IDE. Он позволяет устанавливать точки останова, отслеживать значения переменных и управлять выполнением программы пошагово, что критично при поиске сложных багов.
3. **Поддержка многоплатформенности**, PyCharm работает не только на Windows, но и на **macOS** и **Linux**, что делает его универсальным инструментом для разработчиков, работающих в разных операционных системах;
4. PyCharm поддерживает HTML, CSS, JavaScript, SQL и другие веб-технологии, позволяя редактировать весь проект в одном месте. Особенно полезна возможность подключать базы данных прямо в IDE, выполнять SQL-запросы и просматривать таблицы, не покидая редактора.

Минусы PyCharm:

1. Высокая нагрузка на систему. PyCharm может потреблять значительное количество оперативной памяти и запускаться достаточно долго, особенно на слабых компьютерах. Это не всегда удобно, если нужно быстро внести правку в проект.
2. **Сложности с настройкой для начинающих**. PyCharm предлагает множество настроек и параметров, что может быть полезным для опытных разработчиков, но может запутать новичков. Начинающим пользователям может потребоваться некоторое время, чтобы освоиться в настройках и инструментах IDE.

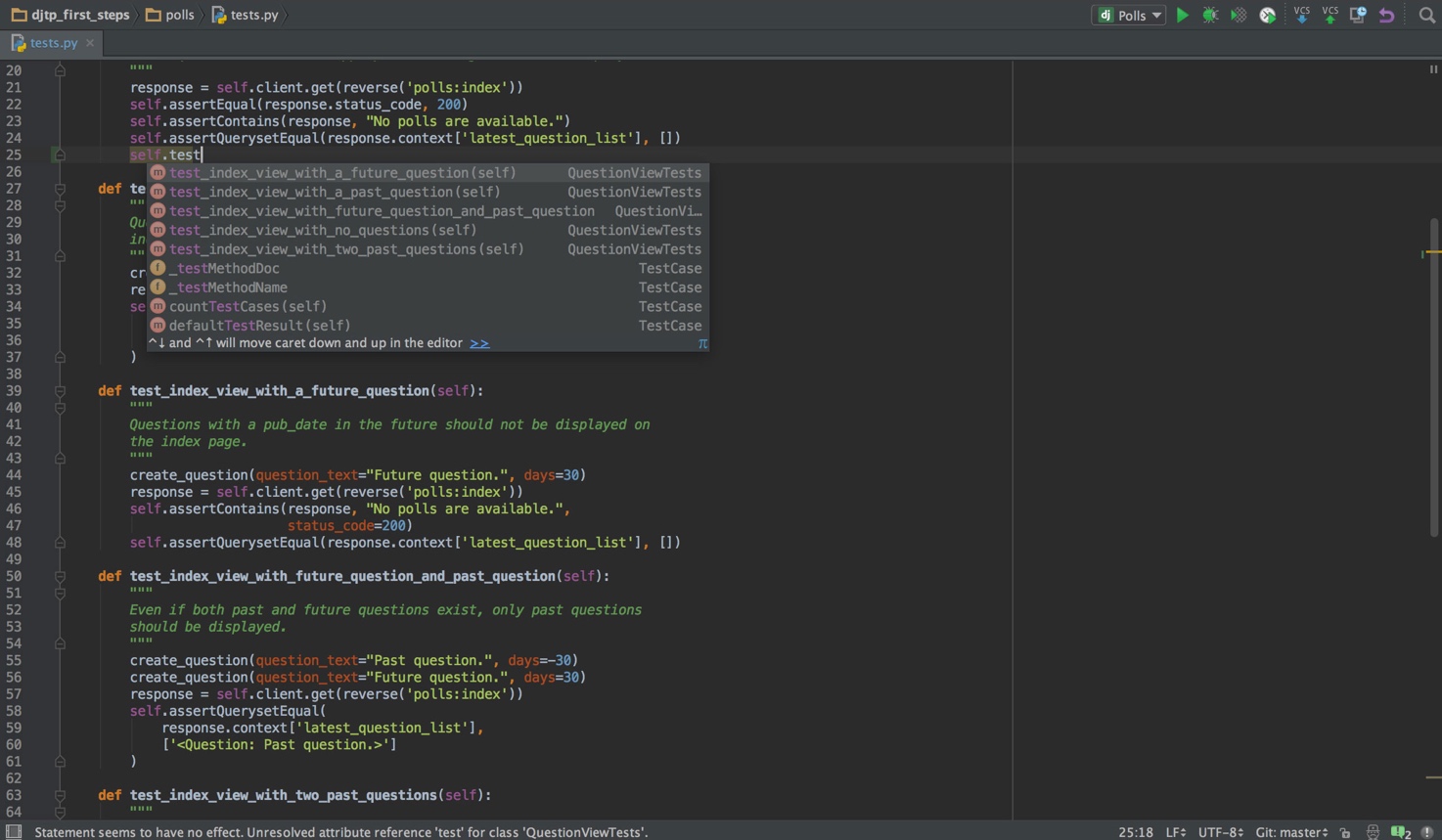
Рисунок 1.8 – IDE Rider

Таблица 1.3 – Сравнительная характеристика интегрированных сред разработки

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Visual Studio** | **Visual Studio Code** | **PyCharm** |
| **Производительность** | + | + | + |
| **Поддержка отладки** | + | \_ | + |
| **Поддержка расширений для Python** | – | + | + |
| **Кроссплатформенность** | – | + | + |
| **Удобство интерфейса** | + | + | + |

В конечном итоге было принято решение выбрать средой разработки PyCharm, т.к. это наиболее привычная среда разработки, с которой имелся длительный опыт работы.

### Система управления базами данных

При выборе СУБД для работы с данными необходимо учитывать следующие ключевые критерии:

1. **Производительность и масштабируемость:** возможность обработки больших объемов данных. Эффективность индексации и оптимизация запросов;
2. **Надежность и отказоустойчивость:** поддержка резервного копирования и восстановления. Гарантия целостности данных;
3. **Совместимость и интеграция:** поддержка разных языков программирования (C#, Java, Python и др.). Интеграция с ORM (Entity Framework, Hibernate). Поддержка REST API и хранимых процедур;
4. **Простота администрирования и мониторинга:** удобные инструменты управления и мониторинга. Гибкость настройки и удобство резервного копирования.

***Microsoft SQL Server (SSMS – SQL Server Management Studio)***

Реляционная СУБД от Microsoft с мощными инструментами управления и администрирования.  
 Плюсы:

1. Высокая производительность и поддержка оптимизации запросов;
2. Полное соответствие ACID;
3. Развитые механизмы безопасности (шифрование, ролевая модель, аудит);
4. Интеграция с .NET, C#, Azure и облачными сервисами Microsoft;
5. Отличные инструменты мониторинга и резервного копирования;
6. Простота настройки и администрирования через SSMS.

**Минусы:**

1. Высокие требования к ресурсам сервера.

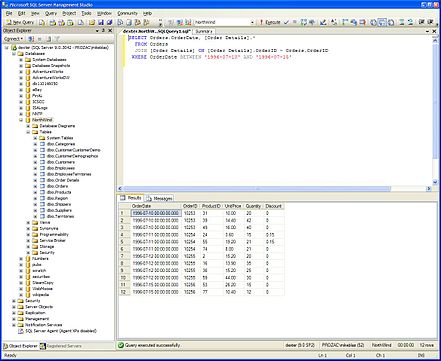


Рисунок 1.9 − СУБД SSMS

***MySQL***

Популярная реляционная СУБД, широко используемая в веб-разработке.  
**Плюсы:**

1. Быстрая обработка запросов;
2. Простота в установке и настойке;
3. Отличная поддержка репликации и кластеризации;
4. Бесплатная версия (Community Edition).

**Минусы:**

1. Ограниченная поддержка ACID (InnoDB – единственный механизм, поддерживающий транзакции);
2. Менее развитые инструменты администрирования по сравнению с SQL Server.

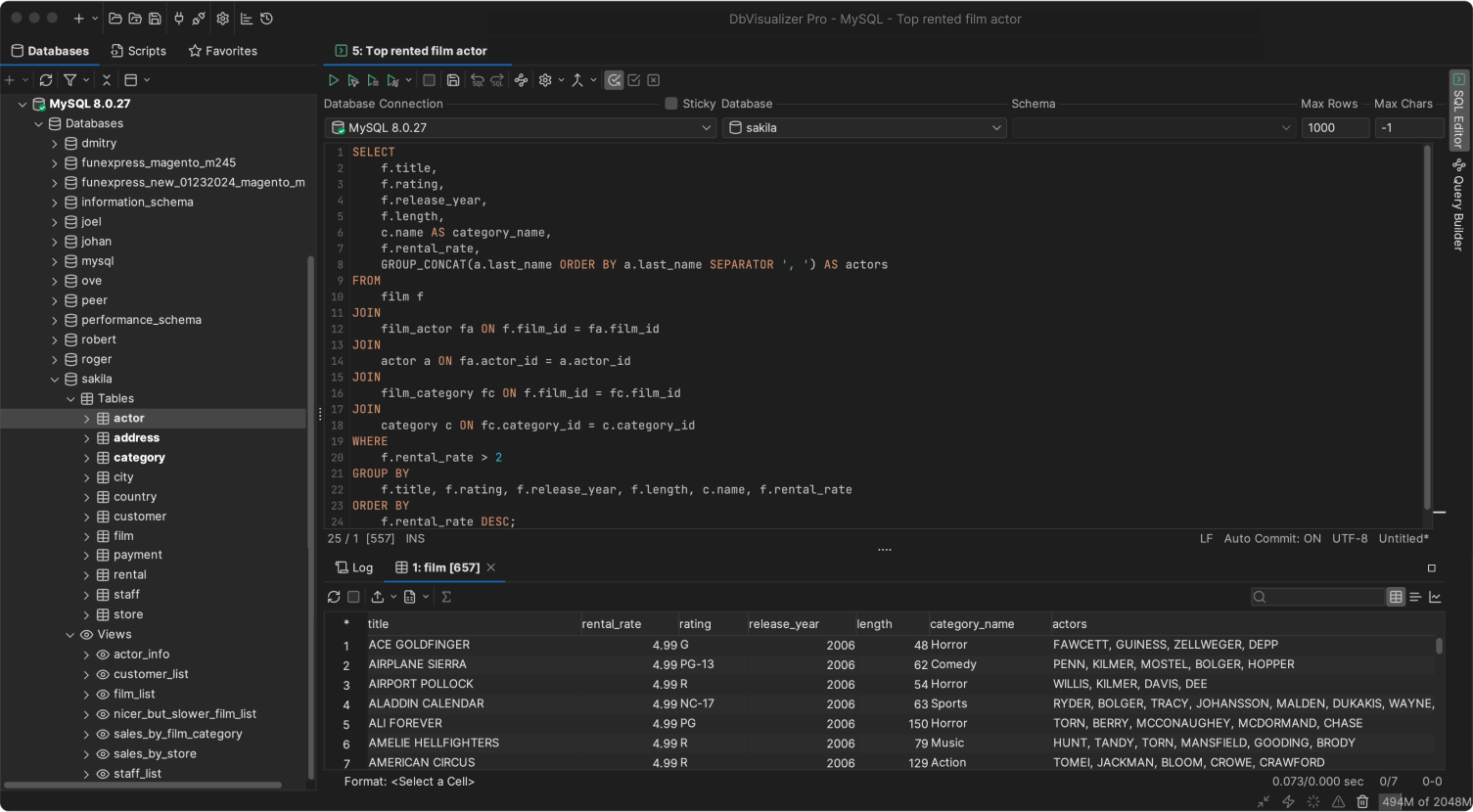


Рисунок 1.10 − СУБД MySQL

***PostgreSQL***

Мощная бесплатная реляционная СУБД с открытым исходным кодом.  
**Плюсы:**

1. Полная поддержка ACID;
2. Хорошая масштабируемость и производительность;
3. Поддержка JSONB, геоданных и расширяемости;
4. Бесплатная и кроссплатформенная.

**Минусы:**

1. Менее удобные инструменты управления;
2. Сложнее в настройке и администрировании по сравнению с MS SQL.

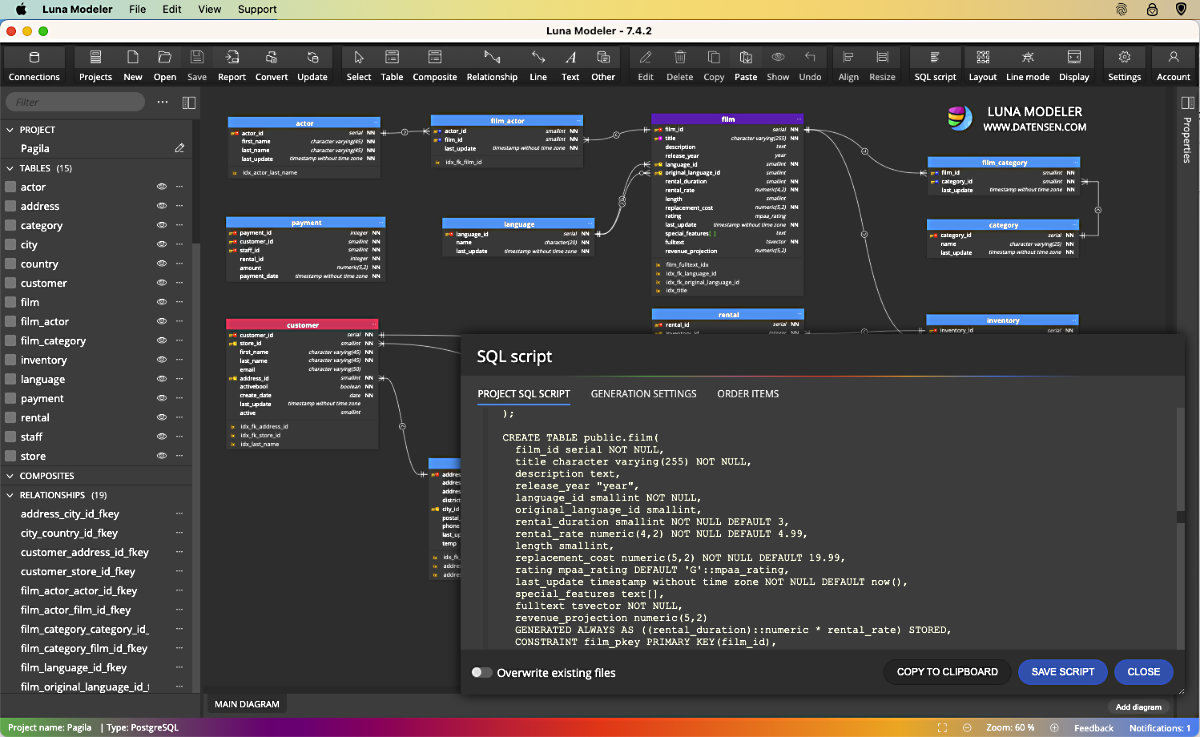


Рисунок 1.11 − СУБД PostgreSQL

Таблица 1.4 – Сравнительная характеристика СУБД

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SSMS** | **MySQL** | **PostgreSQL** |
| **Производительность** | + | − | + |
| **Надежность** | + | + | + |
| **Совместимость** | + | − | + |
| **Простота** | + | + | − |

Проведя сравнение нескольких систем управления базами данных, было принято решение выбрать для разработки **Microsoft SQL Server, из-за личных предпочтений, опыта работы и оптимальной производительности.**

# ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Проектирование и разработка базы данных

Обязательным элементом автоматизации деятельности предприятия является база данных. Она является основным компонентом информационной системы, обеспечивающим надёжное хранение, структурирование и быстрый доступ к информации. Этап проектирования и разработки базы данных включает в себя анализ предметной области, определение требований, построение логической и физической моделей, а также реализацию структуры БД с использованием выбранной СУБД. Ниже рассмотрены все основные этапы, предпринятые в процессе разработки.

### Анализ предметной области

При проектировании базы данных для информационной системы необходимо тщательно изучить предметную область, чтобы создать систему, максимально удовлетворяющую потребности торгового предприятия и пользователей. Анализ предметной области начинается с определения сущностей и их атрибутов.

На основе анализа требований к информационной системе были выделены ключевые сущности, соответствующие основным объектам предметной области:

1. Пользователь (User) – хранит информацию о зарегистрированных пользователях системы. Атрибутами пользователя являются: id, имя пользователя, email, пароль;
2. Продукт (Product) – включает сведения о товарах или услугах, доступных для приобретения. Атрибутами являются: id, название, описание, цена, категория\_id, изображение товара;
3. Категория (Category) – служит для логической группировки продуктов. Атрибутами являются: id, название категории;
4. Заказ (Order) – фиксирует информацию о заказах, созданных пользователями. Атрибутами являются: id, пользователь\_id, дата создания, статус;
5. Состав заказа (OrderItem) – отображает состав заказа с указанием продуктов и их количества. Атрибутами являются: id, заказ\_id, продукт\_id, количество;
6. Оплата (Payment) – хранит данные о проведённых платежах. Атрибутами являются: id, заказ\_id, тип оплаты, статус, дата оплаты.

После определения сущностей установим связи между ними:

1. Один пользователь может оформить несколько заказов (один-ко-многим).
2. Один заказ может содержать несколько товаров, и один товар может входить в состав нескольких заказов (многие-ко-многим).
3. Каждый продукт принадлежит одной категории (многие-к-одному).
4. Каждому заказу соответствует одна запись об оплате (один-к-одному).

#### Определение требований к базе данных

Требования к базе данных можно разделить на две категории: функциональные и нефункциональные.

Функциональные требования описывают конкретные действия, поведение и функции, которые должна выполнять система. Они определяют, что будет делать система, от взаимодействия с пользователем до ответов системы.

Нефункциональные требования, однако, фокусируются на том, насколько хорошо работает система. Они рассматривают такие аспекты, как производительность, безопасность, масштабируемость и надежность, гарантируя, что система не только функционирует так, как ожидается, но и обеспечивает высококачественный пользовательский опыт.

К функциональным требованиям базы данных относится:

1. Хранение информации о товарах, поставщиках, заказах и клиентах – данные должны быть структурированными, доступными для поиска и изменения.
2. Автоматизация процесса оформления заказов – система должна позволять создавать новый заказ, заполнять его необходимыми данными и отслеживать статус выполнения.
3. Формирование отчетов – база данных должна поддерживать возможность формирования отчетов по продажам, движению товаров, эффективности работы поставщиков.
4. Обеспечение безопасности хранения данных – защита информации от несанкционированного доступа, разграничение прав пользователей.

К нефункциональным требованиям мы отнесли возможность масштабируемости базы данных, возможности расширения ее функционала без кардинальных изменений структуры.

#### Построение логической модели базы данных

Рассмотрим, что же такое логическая модель базы данных. Это абстрактное представление структуры данных, которое используется для планирования и проектирования баз данных. Она описывает, как данные будут организованы и как они будут взаимодействовать друг с другом, не привязываясь к конкретной системе управления базами данных (СУБД). Логическая модель помогает разработчикам и аналитикам понять требования к данным и спроектировать эффективную и масштабируемую базу данных.

Логическая модель базы данных является важным этапом в процессе разработки базы данных, так как она позволяет структурировать данные и определить их взаимосвязи. Это помогает избежать ошибок и недоразумений на более поздних этапах разработки. Логическая модель также служит основой для создания физической модели базы данных, которая уже учитывает конкретные особенности выбранной СУБД.

Для успешного построения логической модели базы данных были выделены следующие требования:

1. Найти все необходимые объекты и определите их атрибуты;
2. Выбрать подходящие первичные ключи (PK) в качестве уникальных идентификаторов для групп атрибутов;
3. Нормализовать модель данных в соответствии с эксплуатационными требованиями;
4. Определить отношения между бизнес-объектами в модели данных;
5. Проверить, точно ли объекты данных и их взаимосвязи отражают требуемую логику.

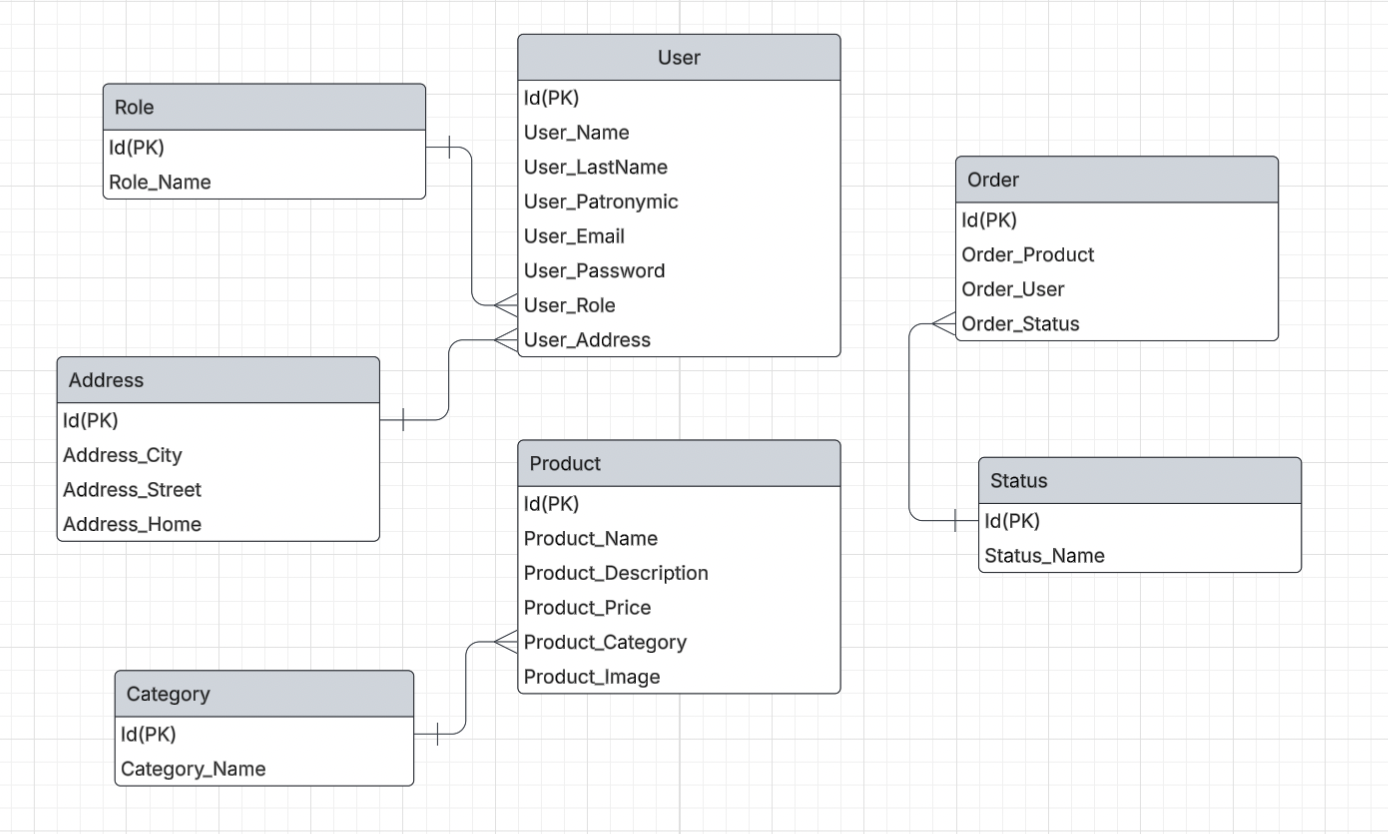
Конечный вариант логической модели базы данных представлен ниже.

Рисунок 2.1 – Логическая модель базы данных

#### Построение физической модели базы данных

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных.

Физические модели данных предоставляют подробные сведения, на основе которых администраторы и разработчики баз данных воспроизводят бизнес-логику в физической базе данных. Эти модели содержат дополнительные атрибуты, которых не было в логической модели данных, например триггеры, хранимые процедуры и типы данных. Поскольку физические модели данных сопоставляют элементы данных с реальной базой данных, в них должны учитываться специфические ограничения платформы, в том числе соглашения об именовании и зарезервированные слова.

Для успешного построения физической модели базы данных были выделены следующие требования:

1. Преобразовать локальную модель данных в соответствии с платформой выбранного поставщика баз данных;
2. Сопоставить объекты данных с таблицами;
3. Определить и создать первичные (PK) и внешние (FK) ключи в таблицах базы данных по мере необходимости;
4. Убедиться, что структура базы данных правильно нормализована, чтобы избежать избыточного дублирования данных и улучшить их целостность;
5. Добавить необходимые ограничения, правила, разделы и программные функции базы данных, которые упростят разработку приложений;
6. Сравнить физическую модель данных и логическую модель данных и убедитесь, что все бизнес-требования правильно преобразованы.

Конечный вариант физической модели базы данных представлен пизже.

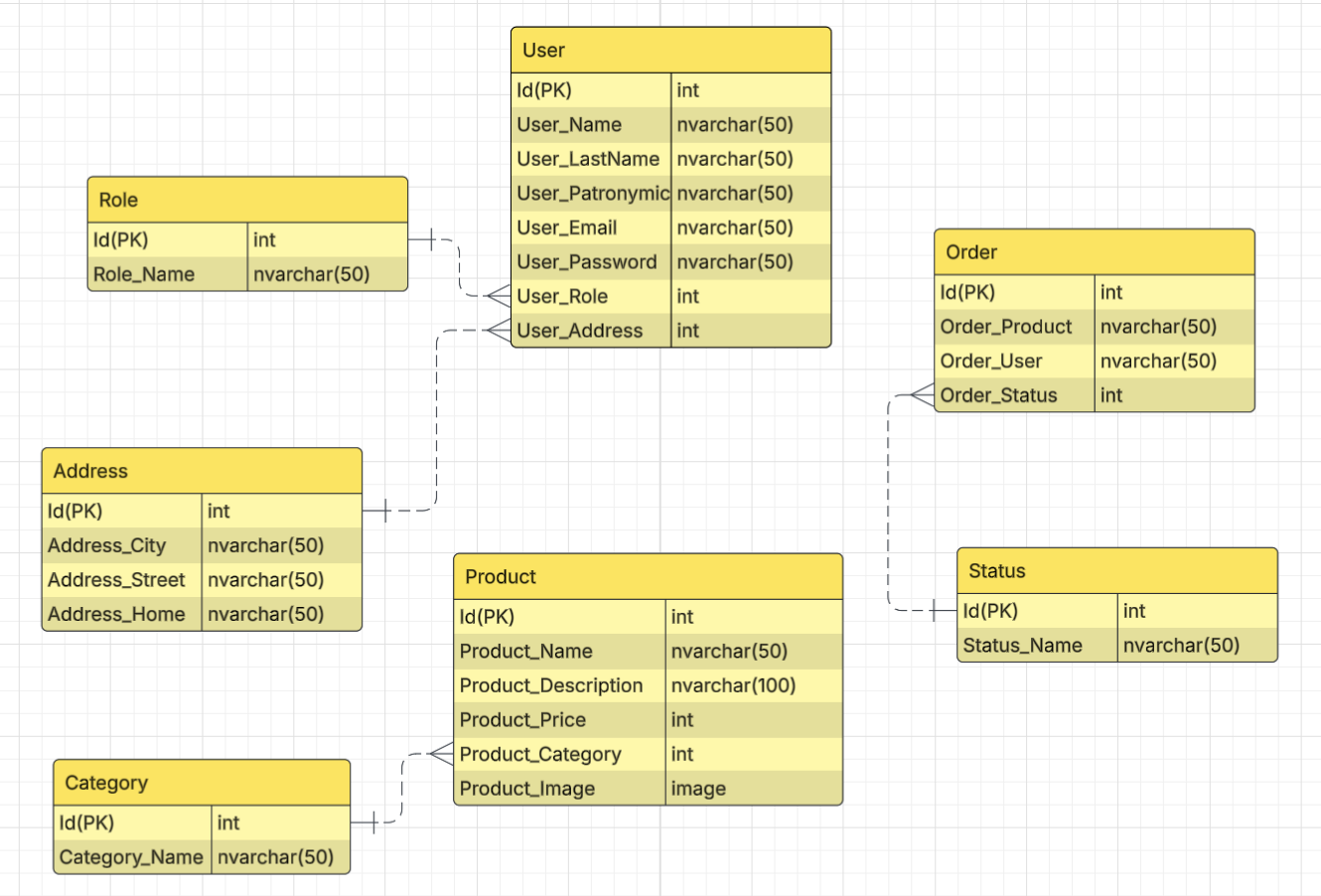


Рисунок 2.2 – Физическая модель базы данных

#### Характеристика базы данных

1. Интеграция – это объединение более мелких компонентов в единую систему, чтобы она могла функционировать как единое целое. [Электронный ресурс] – URL: [https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-data-integration - :~:text=Интеграция — это объединение более мелких,могла функционировать как единое целое.](https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-data-integration#:~:text=%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F%20%E2%80%94%20%D1%8D%D1%82%D0%BE%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B1%D0%BE%D0%BB%D0%B5%D0%B5%20%D0%BC%D0%B5%D0%BB%D0%BA%D0%B8%D1%85,%D0%BC%D0%BE%D0%B3%D0%BB%D0%B0%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D1%82%D1%8C%20%D0%BA%D0%B0%D0%BA%20%D0%B5%D0%B4%D0%B8%D0%BD%D0%BE%D0%B5%20%D1%86%D0%B5%D0%BB%D0%BE%D0%B5.) Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-1)
2. SKU (англ. Stock keeping unit) — это код, состоящий из цифр и букв, который присваивают товарам для отслеживания остатков на складе, подсчета продаж по группам и категориям товаров и для определения оптимального момента для следующих закупок. [Электронный ресурс] – URL: <https://sendpulse.com/ru/support/glossary/sku> Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-2)
3. Маркетплейс (торговая площадка) — это онлайн-платформа для продажи и покупки товаров и услуг через интернет, простыми словами продавцы размещают свои товары, покупатели выбирают лучшие варианты по ценам, характеристикам и другим параметрам. [Электронный ресурс] – URL: <https://kokoc.com/blog/chto-takoe-marketpleysy-prostymi-slovami-kak-oni-rabotayut-plyusy-i-minusy-chem-otlichayutsya-ot-agr/> Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-3)
4. Веб-приложение – это компьютерная программа, которая запускается прямо в веб-браузере. Для работы веб-приложению не требуется установка на компьютер, так что вы можете получить доступ к нему с любого устройства, имеющего веб-браузер, при условии, что оно подключено к сети Интернет.

   [Электронный ресурс] – URL: <https://sibdev.pro/blog/articles/chto-takoe-veb-prilozhenie> Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-4)
5. Фреймворк (англ. framework — «остов, каркас, структура») — готовая модель в IT, заготовка, шаблон для программной платформы, на основе которого можно дописать собственный код. [Электронный ресурс] – URL: <https://secretmag.ru/enciklopediya/chto-takoe-freimvork-obyasnyaem-prostymi-slovami.htm> Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-5)
6. IDE (от англ. Integrated Development Environment, «интегрированная среда разработки») — это программа, в которой разработчики пишут, проверяют, тестируют и запускают код, а также ведут большие проекты. [Электронный ресурс] – URL: <https://practicum.yandex.ru/blog/integrirovannaya-sreda-razrabotki-ide/> Режим доступа: свободный (дата обращения: 12.02.2025). [↑](#footnote-ref-6)