**РЕФЕРАТ**

Разработана система для автоматизации обработки запросов заказов на швейном предприятии. Программно реализованы возможности оформления заказа, добавление записи в уже существующий заказ, поиск по заказам и разработана функция отображения незавершенных заказов. Результатом выполнения программы является база данных, в которой находятся данные о всех принятых заказах. Также был проведен всесторонний анализ этапов, сопутствующих разработке и внедрению в жизнь данного проекта.

**ANNOTATION**

A system for the automation of request processing orders for the clothing company was developed. The software features implemented an order, add records to an existing order, search orders and developed a display function incomplete orders. The result of the program is a database that contains data on all accepted orders. Also, a comprehensive analysis of the stages that accompany the development and implementation in the life of the project.

СОДЕРЖАНИЕ

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 7](#_Toc317085981)

[1. ВВЕДЕНИЕ 8](#_Toc317085982)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 11](#_Toc317085983)

[2. ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ АВТОМАТИЗАЦИИ ДОКУМЕНТООБОРОТА 13](#_Toc317085984)

[2.1 1С:Предприятие 8 13](#_Toc317085985)

[2.1.1 1С:Управление небольшой фирмой 8 15](#_Toc317085986)

[2.2 СЭД DOCUMETR 20](#_Toc317085987)

[2.3 Платформа DocsVision 27](#_Toc317085988)

[3 ЯЗЫК PHP 31](#_Toc317085989)

[3.1 Возможности PHP 32](#_Toc317085990)

[3.2 Преимущества PHP 35](#_Toc317085991)

[3.3 История развития PHP 39](#_Toc317085992)

[4 ОСНОВНАЯ ИНФОРМАЦИЯ О БАЗАХ ДАННЫХ 54](#_Toc317085993)

[4.1 MySQL 58](#_Toc317085994)

[5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 64](#_Toc317085995)

[5.1 Концепция экономического обоснования научно-технической разработки 64](#_Toc317085996)

[5.2 Потребительские свойства продукта, связанного с совершенствованием продукта (технологии) 65](#_Toc317085997)

[5.3 Рынок и план маркетинга 66](#_Toc317085998)

[5.4 Производство продукции 69](#_Toc317085999)

[5.5 Организационный план проекта 72](#_Toc317086000)

[5.6 Экономическая эффективность разработки 74](#_Toc317086001)

[5.7 Выводы 78](#_Toc317086002)

[6 ОХРАНА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ 79](#_Toc317086003)

[6.1 Программа для ЭВМ как объект правовой охраны 79](#_Toc317086004)

[6.2 Официальная регистрация программ для ЭВМ 81](#_Toc317086005)

[6.3. Коммерческая реализация программ для ЭВМ 86](#_Toc317086006)

[ЗАКЛЮЧЕНИЕ 91](#_Toc317086007)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 92](#_Toc317086008)

[ПРИЛОЖЕНИЕ 1 93](#_Toc317086009)

[Исходный текст модуля Index.php 93](#_Toc317086010)

[Исходный текст модуля Orders.php 98](#_Toc317086011)

[Исходный текст модуля Search.php 99](#_Toc317086012)

[Исходный текст модуля Order.sql 103](#_Toc317086013)

[ПРИЛОЖЕНИЕ 2 105](#_Toc317086014)

# ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных

НИР – научно-исследовательская работа

СУБД – система управления базами данных

СЭД – система автоматизации документооборота

СЭДО – система электронного документооборота

# ВВЕДЕНИЕ

Система автоматизации документооборота, система электронного документооборота (СЭДО) — автоматизированная многопользовательская система, сопровождающая процесс управления работой иерархической организации с целью обеспечения выполнения этой организацией своих функций. При этом предполагается, что процесс управления опирается на человекочитаемые документы, содержащие инструкции для сотрудников организации, необходимые к исполнению.

Основные принципы электронного документооборота:

1. Однократная регистрация документа, позволяющая однозначно идентифицировать документ;
2. Возможность параллельного выполнения операций, позволяющая сократить время движения документов и повышения оперативности их исполнения
3. Непрерывность движения документа, позволяющая идентифицировать ответственного за исполнение документа (задачи) в каждый момент времени жизни документа (процесса);
4. Единая (или согласованная распределённая) база документной информации, позволяющая исключить возможность дублирования документов;
5. Эффективно организованная система поиска документа, позволяющая находить документ, обладая минимальной информацией о нём;
6. Развитая система отчётности по различным статусам и атрибутам документов, позволяющая контролировать движение документов по процессам документооборота и принимать управленческие решения, основываясь на данных из отчётов;

Классификация систем электронного документооборота:

1. Универсальные «коробочные» СЭДО:

1.1 стандартный набор функций;

1.2 невозможность полного соответствия потребностям конкретной организации;

1.3 низкие временные затраты на приобретение и установку;

1.4 относительно низкая стоимость;

1.5 необходимость приобретения лицензии на каждое внедряемое рабочее место.

2. Индивидуально разрабатываемые СЭДО:

2.1 максимально персонифицированная система;

2.2 большие временные затраты;

2.3 высокая стоимость разработки;

2.4 сопутствующие расходы: затраты на обучение сотрудников, покупку нового оборудования и программного обеспечения.

3. Комбинированные СЭДО:

3.1 базовая платформа, к которой разрабатываются необходимые дополнительные модули;

3.2 полное соответствие нуждам предприятия;

3.3 небольшие временные затраты на разработку и внедрение;

3.4 стоимость включает: цену базовой платформы и стоимость индивидуальной доработки, зависящей от сложности заказа;

3.5 передача заказчику прав на продукт;

3.6 простота освоения и использования;

3.7 полная русификация;

3.8 удобный интерфейс;

3.9 взаимодействие с существующими офисными приложениями. [1]

Описание задачи:

Прием заказов – это один из важнейших процессов на предприятиях, преимущественно занимающихся оказанием услуг или торговлей. Именно здесь зарождаются будущие отношения с клиентами, которые в перспективе приносят организации прибыль.

Под приемом заказов здесь подразумевается получение информации от клиента о нужной ему услуге или продукте. В современных условиях ведения бизнеса, важно не только быстро и корректно принять заказ от клиента, но еще и иметь возможность отследить судьбу заказа, а значит, все действия персонала по приему заказов должны быть достаточно прозрачны.

Какие функции должно выполнять, и какие задачи должно решать подразделение по приему заказов на предприятии? Прежде всего – это:

1. Регистрация обращений клиентов, сохранение истории обращений.
2. Быстрое и корректное заполнение бланков заказов (а значит, появляется необходимость систематизации и автоматизации деятельности).
3. Эффективное взаимодействие всех сотрудников подразделения по приему и реализации заказов.
4. Контроль этапов жизненного цикла заказа (прием, обработка и изготовление/поставка продукции).
5. Высокая степень защиты информации.

В результате выполнения дипломного проекта разработана программная система для автоматизации обработки запросов заказов на швейном предприятии.

В ходе изучения материала интересующей нас области проанализированы различные системы автоматизации документооборота. Были рассмотрены возможности и преимущества языка PHP и базы данных MySQL. Далее представлено описание системы, созданной в рамках данного проекта, приведён пример её использования. Также приведён раздел по технико-экономическому обоснованию, где был посчитан экономический эффект проекта в виде чистой текущей стоимости проекта в размере 23901 рублей, и раздел, посвящённый охране интеллектуальной собственности.

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Темой данного дипломного проекта является разработка системы для автоматизации обработки запросов заказов на швейном предприятии.

Описание задачи:

Прием заказов – это один из важнейших процессов на предприятиях, преимущественно занимающихся оказанием услуг или торговлей. Именно здесь зарождаются будущие отношения с клиентами, которые в перспективе приносят организации прибыль.

Под приемом заказов здесь подразумевается получение информации от клиента о нужной ему услуге или продукте. В современных условиях ведения бизнеса, важно не только быстро и корректно принять заказ от клиента, но еще и иметь возможность отследить судьбу заказа, а значит, все действия персонала по приему заказов должны быть достаточно прозрачны.

Какие функции должно выполнять, и какие задачи должно решать подразделение по приему заказов на предприятии? Прежде всего – это:

1. Регистрация обращений клиентов, сохранение истории обращений.
2. Быстрое и корректное заполнение бланков заказов (а значит, появляется необходимость систематизации и автоматизации деятельности).
3. Эффективное взаимодействие всех сотрудников подразделения по приему и реализации заказов.
4. Контроль этапов жизненного цикла заказа (прием, обработка и изготовление/поставка продукции).
5. Высокая степень защиты информации.

Содержание работы:

1. Введение
2. Обзор существующих систем автоматизации документооборота
3. Язык PHP
4. Основная информация о базах данных
5. Технико-экономическое обоснование проекта
6. Охрана интеллектуальной собственности

Заключение

Список используемой литературы

Технические требования: программный продукт, автоматизирующий обработку запросов заказов на швейном предприятии , должен выполняться на компьютере типа: компьютер (десктоп); процессор: Intel Pentium G6950; 2,8 GHz; оперативная память: 1024 MB; PC3-10666; жесткий диск: 100 GB; графический адаптер: Intel GMA HD; 512 M.

# ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ АВТОМАТИЗАЦИИ ДОКУМЕНТООБОРОТА

## 1С:Предприятие 8

Система программ «1С:Предприятие 8» включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.

Области применения.

Гибкость платформы позволяет применять 1С:Предприятие 8 в самых разнообразных областях:

1. автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.
2. поддержка оперативного управления предприятием;
3. автоматизация организационной и хозяйственной деятельности;
4. ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность;
5. широкие возможности для управленческого учета и построения аналитической отчетности, поддержка многовалютного учета;
6. решение задач планирования, бюджетирования и финансового анализа;
7. расчет зарплаты и управление персоналом;
8. другие области применения.

Технологическая платформа.

Платформа 1С:Предприятие 8 была создана с учетом 6-летнего опыта применения системы программ 1С:Предприятие 7.7, которую используют десятки тысяч разработчиков. Несмотря на значительные изменения, новая версия 8 сохранила идеологическую преемственность с предыдущими версиями.

Прикладные решения.

Фирма "1С" выпускает тиражные прикладные решения, предназначенные для автоматизации типовых задач учета и управления в коммерческих предприятиях реального сектора и бюджетных организациях. В каждом программном продукте сочетается использование стандартных решений (общих для всех или нескольких программ) и максимальный учет специфики задачи конкретной отрасли или рода деятельности предприятия.

Отраслевые и региональные прикладные решения создаются силами партнеров-разработчиков и предназначены для автоматизации отдельных направлений или областей деятельности предприятий. Все они сертифицированы на требования "1С:Совместимо".

1C:Библиотека стандартных подсистем.

Инструмент разработчика "1С:Библиотека стандартных подсистем 8.2" (БСП) предоставляет набор универсальных функциональных подсистем и технологию для разработки прикладных решений на платформе "1С:Предприятие 8.2". С помощью БСП можно быстро создавать новые конфигурации с уже готовой базовой функциональностью, а также включать готовые функциональные блоки в существующие конфигурации. Использование БСП при разработке прикладных решений позволит также достичь большей стандартизации конфигураций, что уменьшит время на изучение и внедрение прикладных решений за счет их унификации по набору используемых стандартных подсистем.

Внедрения.

Внедрения выполняются силами партнеров-внедренцев и реализуют особенности деятельности конкретного предприятия или специальные пожелания заказчика.

Внедрения и адаптации прикладных решений также могут выполняться и силами IT-специалистов заказчика, самостоятельно, или во взаимодействии с партнерами-внедренцами.

Внедрение корпоративных информационных систем на платформе 1С:Предприятие 8.

Опыт внедрения прикладных решений на платформе 1С:Предприятие 8 показывает, что система позволяет решать задачи различной степени сложности - от автоматизации одного рабочего места до создания информационных систем масштаба предприятия.

В то же время, внедрение большой информационной системы предъявляет повышенные требования по сравнению с небольшим или средним внедрением. Информационная система масштаба предприятия должна обеспечивать приемлемую производительность в условиях одновременной и интенсивной работы большого количества пользователей, которые используют одни и те же информационные и аппаратные ресурсы в конкурентном режиме.[2]

### 1С:Управление небольшой фирмой 8

"1С:Управление небольшой фирмой 8" (УНФ) – это готовое решение для автоматизации оперативного управления на предприятиях малого бизнеса.

В программе реализовано все самое необходимое для ведения оперативного учета, контроля, анализа и планирования на предприятии. Решение не перегружено излишним функционалом, его можно легко настроить на особенности организации управления и учета в компании – это обеспечивает возможность "быстрого старта" и удобство ежедневной работы.

УНФ помогает повысить эффективность работы компании, предоставляя владельцам и руководителям широкий спектр инструментов для управления, а сотрудникам – новые возможности для продуктивной ежедневной работы.

Одна программа заменяет десятки таблиц, документов и отчетов на вашем столе и на компьютере – все данные регистрируются и хранятся в единой информационной базе

Все данные – в единой информационной базе

В программе "1С:Управление небольшой фирмой 8" регистрируются как уже совершенные, так и планируемые хозяйственные операции и события. Например, обязательства перед клиентами, заказы покупателей, состояние заказов, задания сотрудников, планируемая загрузка ресурсов предприятия, планы-графики выполнения работ, производства, планы продаж и много другое.

В единой информационной базе:

• база клиентов,

• банковские и кассовые операции, клиент-банк, платежный календарь,

• расчеты с контрагентами, персоналом,

• учет материалов, товаров, продукции,

• заказы клиентов, заказы-наряды,

• планирование и учет выполнения работ и оказания услуг,

• планирование и учет производственных операций,

• планирование загрузки ресурсов предприятия,

• торговые операции, в том числе розничные продажи,

• учет персонала, расчет управленческой заработной платы,

• учет затрат и расчет себестоимости,

• имущество, капитал,

• доходы, расходы, прибыли и убытки

• финансовое планирование (бюджетирование) и т. д.

В программе предусмотрено оформление практически всех первичных документов торгового, складского и производственного учета, а также документов движения денежных средств.

Широкий спектр отчетов обеспечивает владельцам, руководителям и сотрудникам возможность быстро получать информацию – в удобной для работы и принятия решений форме, с требуемой оперативностью и детализацией.

Программа не предназначена для ведения бухгалтерского и налогового учета – для этих целей можно использовать "1С:Бухгалтерию 8", в которую автоматически передается необходимая информация из УНФ.

Программу можно использовать для нескольких компаний или частных предпринимателей – как независимых, так и работающих в рамках одного бизнеса. При изменении масштабов и структуры бизнеса, подходов к управлению или организации работ программа может быть перенастроена без больших затрат времени и денег.

Рекомендуемое число рабочих мест – от одного до десяти.

Преимущества, которые дает программа владельцам и руководителям

Реальная польза от автоматизации – уже через несколько дней или недель."1С:Управление небольшой фирмой 8" – это готовое решение, которое можно быстро запустить в эксплуатацию, а при изменении масштабов бизнеса, подходов к управлению или организации работ перенастроить без больших затрат времени и денег.

Одна программа заменяет десятки таблиц, документов и отчетов на вашем столе и на компьютере – все данные регистрируются и хранятся в единой информационной базе. Программа помнит все – каждого покупателя, цены, скидки, заказы, обязательства, договоры.

Информация для принятия решений – по одному нажатию мышкой.

• "Движение денежных средств",

• "Платежный календарь",

• "Доходы и расходы, прибыли и убытки",

• "Расчеты с дебиторами/кредиторами",

• "Состояние заказов",

• "Динамика и анализ продаж",

• "План-график работ" и "Загрузка ресурсов",

• другие управленческие отчеты.

Все под контролем – ведение графиков выполнения работ и производства, загрузки ресурсов, отгрузок и поставок товаров, нарядов и заданий сотрудников обеспечивает оперативный контроль деятельности.

Широкие возможности для планирования деятельности – финансовое планирование (бюджетирование), планирование продаж, загрузки персонала, загрузки ресурсов предприятия и пр.

Труд руководителей и сотрудников становится более производительным – за счет автоматизации рутинных операций, быстрой и удобной подготовки информации для принятия решений на разных уровнях. Современный эргономичный интерфейс обеспечивает легкость освоения для начинающих и высокую скорость работы для опытных пользователей.

Ваших знаний достаточно – чтобы использовать программу, не требуется знать бухгалтерский и налоговый учет. Программа не перегружена излишними возможностями, в ней реализовано все только самое необходимое для учета, контроля, анализа и планирования в малом бизнесе.

Доступ к информации из любой точки земного шара – через Интернет

Области применения:

"1С:Управление небольшой фирмой 8" рекомендуется использовать для автоматизации управленческого учета в сервисных, производственных и торговых компаниях малого бизнеса. Например, в компаниях следующих направлений деятельности:

• аренда,

• автосервис,

• издательский бизнес,

• инжиниринг,

• интернет-торговля,

• информационные технологии,

• кадровые агентства,

• консалтинг,

• наука и научное обслуживание,

• торговля,

• обучение,

• охранные услуги,

• посреднические услуги,

• поставки товаров, продукции, материалов и оборудования,

• подрядные (проектные) работы,

• производство,

• реклама и дизайн,

• риэлтерские услуги,

• сфера бытовых услуг (ателье, салоны красоты, прачечные, уборка помещений и пр.),

• сервисные компании,

• центры технического обслуживания,

• строительство и ремонт,

• транспортные услуги,

• туризм и экскурсионное обслуживание,

• юридические консультации и нотариальные услуги. [3]

## СЭД DOCUMETR

DOCUMETR® for SharePoint - мощная система электронного документооборота на базе Microsoft SharePoint. Включает все необходимые функции по совместной работе с документами и заданиями.

Данная программа электронного документооборота на предприятиипозволяет автоматизировать любые бизнес-процессы, связанные с движением документов и исполнением заданий.

Система электронного документооборота DOCUMETR® успешно подходит как для крупных предприятий с численностью более 500 сотрудников, так и для небольших компаний (при этом решение базируется на бесплатной платформе Windows SharePoint Services 3.0).

Функциональные возможности программы автоматизации документооборота DOCUMETR:

1. Обработка входящей и исходящей корреспонденции (обращения в службу поддержки, call-центр, факсы, письма, телефонограммы и др.).
2. Организация внутреннего документооборота (приказы, распоряжения, служебные записки, заявки, договора и т.п.).
3. Контроль исполнения поручений (постановка задач, согласование и контроль их исполнения). У каждого пользователя есть домашняя страница, на которой отображаются задания, сгруппированные по типам: «Актуальные задания», «Инициированные задания», «Контролируемые задания», «Завершенные задания».
4. Создание документов на основе шаблонов (шаблоны можно создавать самостоятельно и сохранять их в библиотеке шаблонов).
5. Гибкая настройка маршрутов движения заданий и документов (используя встроенный язык описания маршрутов на базе XML можно выполнить создание нового контура системы и его настройку без программирования).
6. Ведение корпоративной библиотеки (учет печатных и электронных изданий, возможность группировки по типам документов).
7. Ведение полной истории выполнения заданий и изменения документов (при наличии соответствующих прав доступа пользователь может посмотреть всю историю по интересующему документу).
8. Удобное интегрированное средство сканирования документов (например, входящие письма).
9. Удобный поиск информации (сквозной полнотекстовый поиск по карточкам заданий и документам).

Система DOCUMETR® представляет собой расширение для Microsoft SharePoint и позволяет организовать систему электронного документооборота (СЭД).

DOCUMETR® - программа управления электронными документами. В системе заложена универсальная идеология, базирующаяся на следующих основных принципах:

1. В системе циркулируют задания.
2. Каждое задание описывается карточкой задания. Карточка задания содержит несколько полей (атрибутов задания). Также в карточке задания могут присутствовать вложенные документы.
3. Для каждого задания назначаются права доступа. Права доступа определяют возможность доступа разных пользователей к различным атрибутам (полям) карточки задания, а также к заданию в целом.
4. Задания группируются по типам. Тип задания (например, “Входящие письма” или “Договоры”) определяет, какие поля присутствуют в карточке задания, а также определяет права доступа пользователей к заданию и маршрут движения задания в системе. Ключевая особенность системы в том, что кроме встроенных типов заданий можно создавать любые собственные типы заданий с использованием механизма наследования.

Каждое задание имеет жизненный цикл из нескольких этапов. Каждое задание проходит несколько этапов от этапа «Начало» до этапа «Завершено». Этап задания является одним из атрибутов карточки задания. Ключевая особенность системы в том, что можно самостоятельно создавать произвольное количество этапов и гибко настраивать переходы между ними.

Переходы между этапами определяются маршрутом. Маршрут задается индивидуально для каждого типа задания и определяет, какие этапы проходит задание, при каких условиях выполняется переход между этапами, доступные на каждом этапе атрибуты задания и какие пользователи имеют права доступа к этим атрибутам.

Переход между этапами осуществляется при выполнении действия. Перевод задания с одного этапа на другой выполняется либо пользователем системы либо самой системой (в зависимости от определенных условий и событий, например, при завершении запланированного срока на согласование документа).

Система DOCUMETR® имеет ряд особенностей, которые делают ее достаточно удобной при реализации масштабируемой системы электронного документооборота на предприятии:

1. Можно создавать собственные типы заданий путем наследования от существующих. Это представляет фактически неограниченную возможность расширения функциональных контуров в системе электронного документооборота (даже после того как она внедрена и активно используется).
2. Маршруты гибко задаются с использованием XML-описания. В маршрутах можно задавать любые права доступа к элементам заданий, можно задавать значения по умолчанию для атрибутов заданий, можно отдельно задавать права на документы, прикрепленные к заданию. Маршруты очень просто создавать. Пример описания этапа задания в маршруте:

<Начало Название="Постановка задачи">  
<Доступ Запись="Руководство компании; Назначена" />  
<Доступ Чтение="Все заинтересованные" />  
<Запись Поле="Название задачи" />  
<Запись Поле="Описание" />  
<Запись Поле="Ответственный" Умолчание ="Текущий" Обязательное="Да"/>  
<Запись Поле="Срок" Умолчание="Текущий" />  
<Запись Поле="Проект" />  
<Запись Поле="Ссылка на основание" />  
<Маршрут Действие="Передать на выполнение" СледующийЭтап="Выполнение" Назначена="Ответственный">  
<Установка Поле="Дата начала" Значение="Текущий"/>

1. Поддерживаются условные переходы в маршрутах. В маршрутах правила перехода заданий с одного этапа на другой этап задаются не только в виде статических последовательностей, но также и в виде переходов в зависимости от значений конкретных атрибутов (например, “если сумма договора меньше 100 тыс. руб., то он утверждается финансовым директором, иначе – генеральным директором”).
2. Поддерживаются функции «Замещение» и «Делегирование». При использовании замещения или делегирования одни пользователи в системе могут выполнять задания или согласовывать документы за других.
3. Поддерживается параллельное и последовательное согласование. Система поддерживает параллельное согласование в рамках одного этапа и последовательное согласование в рамках нескольких этапов.
4. Поддерживается функциональность сканирования документов. В системе реализована возможность сканирования документов непосредственно из окна браузера с прямым вложением результатов сканирования в карточку задания.
5. Поддерживается связь с проектной моделью. В карточках задания можно указывать проекты, к которым относятся те или иные задания. При этом в процессе движения документа или задания в системе, маршрут движения будет рассчитываться исходя из того, кто является руководителем проекта.
6. Поддерживается экспорт данных и формирование отчетов. Система поддерживает возможность экспорта данных и формирования отчетов. [4]

Продукты линейки DOCUMETR for SharePoint:

Базовые продукты:

1) Платформа DOCUMETR for SharePoint – корпоративный портал + система электронного документооборота в одном решении. DOCUMETR for SharePoint – это базовая платформа (движок) для построения произвольных конфигураций корпоративных порталов и систем документооборота по индивидуальным требованиям заказчиков. Продукт представляет собой расширение для Microsoft SharePoint и содержит универсальный механизм создания произвольных бизнес-процессов с использованием настройки в формате XML-описаний (без программирования!). В состав данного продукта не входит никаких прикладных бизнес-процессов (кроме 2-х демонстрационных контуров «Задачи» и «Входящая корреспонденция»). Благодаря тому, что система построена на базе Microsoft SharePoint, она одинаково успешно подходит как для корпоративных порталов, так и систем

электронного документооборота. Данный продукт ориентирован на следующие группы потребителей:

a. Компании, планирующие полностью самостоятельно внедрять и сопровождать систему электронного документооборота и/или корпоративный портал.

b. Компании, имеющие специфические требования к построению бизнес-процессов (когда готовые бизнес-процессы из состава Пакетных продуктов DOCUMETR не подходят).

Цены:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Лицензия на 20 пользователей и 6 бизнес-  процессов | Лицензия на 50 пользователей  и 12 бизнес-  процессов | Лицензия на 100 пользователей и 12 бизнес-  процессов | Лицензия без  ограничения числа пользователей и количества бизнес-процессов |
| Платформа  DOCUMETR  For SharePoint\* | 14529 руб. | 40681 руб. | 61022 руб. | 130761 руб. |

\* - в состав пакета входит непосредственно лицензия на использование программного обеспечения, установочный пакет программного обеспечения в электронном виде, эксплуатационная документация в электронном виде, удаленная техническая поддержка в объеме 20 часов, право бесплатного получения обновленных версий в течение 1 года с момента приобретения лицензии.

Пакетные продукты (включают в себя платформу DOCUMETR for SharePoint + пакет функциональных расширений):

1) DOCUMETR Lite – готовое решение для малых компаний, включающее базовый набор функциональных расширений. Данное решение поставляется с предварительно настроенными бизнес-процессами (5 шт.), обеспечивающими базовые потребности малых компаний. Все предварительно настроенные бизнес-процессы могут быть гибко изменены пользователем под свои индивидуальные потребности. Максимальное число бизнес-процессов в данной версии – 6.

2) DOCUMETR Basic – готовое решение для применения большинством компаний; содержит типовой набор функциональных расширений, применяемых для автоматизации деятельности большинством организаций. Данное решение поставляется с предварительно настроенными бизнес-процессами (10 шт.), обеспечивающими типовые потребности большинства компаний. Все предварительно настроенные бизнес-процессы могут быть гибко изменены пользователем под свои индивидуальные потребности. Максимальное число бизнес-процессов в данной версии – 12.

3) DOCUMETR Full – готовое решение для корпоративных заказчиков. Полная версия системы, включающая все имеющиеся функциональные расширения (бизнес-процессы) и не имеющая ограничений на количество создаваемых бизнес-процессов.

Цены:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Лицензия на 20 пользователей | Лицензия на 50 пользователей | Лицензия на 100 пользователей | Лицензия без ограничения числа пользователей |
| DOCUMETR  Lite (до 6 бизнес-процессов) | 20341 руб. | 40681 руб. | 61021 руб. | 101703 руб. |
| DOCUMETR  Basic (до 12  бизнес-  процессов) | 26152 руб. | 52304 руб. | 78456 руб. | 130761 руб. |
| DOCUMETR  Full (без  ограничения  числа бизнес-  процессов) | 31963 руб. | 63927 руб. | 95891 руб. | 159819 руб. |

\* - в состав пакета входит непосредственно лицензия на использование программного обеспечения, установочный пакет программного обеспечения в электронном виде, эксплуатационная документация в электронном виде, удаленная техническая поддержка в объеме 40 часов, право бесплатного получения обновленных версий в течение 2 лет с момента приобретения лицензии. [5]

## Платформа DocsVision

Название продукта: Платформа DocsVision

Категория продукта: Платформа

Канал продажи: DocsVision

Техподдержка: DocsVision

Описание:

Платформа DocsVision является базисом для электронного документооборота - создания системы управления документами и бизнес-процессами и включает в себя только основные элементы, предназначенные для хранения информации, доступа к ней, а также обеспечивающие пользователю привычный и удобный интерфейс. Платформа состоит из клиентской и серверной частей.

Серверные компоненты позволяют создать все необходимые структуры для хранения данных (учетных карточек и файлов электронных документов) а также поддерживать их целостность в дальнейшем, обеспечивают получение, обновление и удаление данных клиентом в соответствии с правами доступа, а также многие другие функции.   
Клиентские компоненты предоставляют конечному пользователю удобный и понятный интерфейс для осуществления всей работы с информацией, содержащейся в базе данных, включая ее поиск, просмотр и модификацию.

Основным клиентским компонентом платформы DocsVision является Навигатор, предоставляющий доступ ко всем объектам DocsVision: просмотр дерева папок и карточек, находящихся в папках; просмотр содержимого карточек и их группировку по определенным признакам; поиск карточек по заданным критериям, а также создание новых и редактирование уже существующих документов. Навигатор работает как приложение в среде Microsoft Internet Explorer.

Для наглядности и удобства работы с электронными документами карточки сгруппированы в Папки, аналогичные папкам Проводника, обеспечивающие логичное и упорядоченное хранение документов согласно установленным в организации правилам.

Формы для регистрации документов — карточки документов — могут отличаться в каждой отдельной организации и разрабатываться ею самостоятельно, поэтому они не входят в состав платформы DocsVision, а являются частью встроенного приложения DocsVision «[Делопроизводство](http://docsvision.com/nc/catalogue/dv/dv_20.html)».

Все приложения, создаваемые с использованием DocsVision обеспечивают:

1. Единые механизмы хранения и представления информации
2. Единые механизмы разграничения прав доступа к электронным документам и действиям с ними
3. Общие средства навигации и доступа к данным и организации рабочего пространства пользователя
4. Общую структуру справочников
5. Общие средства поиска и извлечения знаний
6. Единые средства управления жизненным циклом приложений и маршрутизации
7. Единую инфраструктуру управления и администрирования приложений
8. Единые средства разработки (настройки) инфраструктуры
9. Единообразная объектная модель для всех приложений системы и наличие общего набора компонентов, используемых для встраивания в клиентские приложения.

В концепцию системы положена идеология компонентов, неориентированных на соединение с сервером, что обеспечивает максимальные возможности масштабирования системы.

Для выполнения этих требований система реализуется в трехзвенной архитектуре, средним звеном которой выступает Web-сервер. На Web-сервере работает один из основных компонентов ядра - сервер хранилища, который обеспечивает доступ к данным и хранилищу файлов. Сервер хранилища реализуется как stateless-компонент, что позволяет решить проблему масштабируемости на уровне среднего звена приложения.

Кроме того, при помощи Web-сервера решается проблема установки модулей на клиентской рабочей станции, что резко снижает затраты на установку и обновление системы.

Система позволяет работать одновременно с несколькими серверами баз данных, а также может обеспечивать балансировку загрузки серверов приложений DocsVision, при работе под управлением Microsoft AppCenter.

DocsVision позволит обеспечивать работу с:

1. Электронными документами, содержащие большие объемы данных
2. Большими количествами документов, хранящимися на сервере
3. Большим количеством одновременных пользовательских сессий

Архитектура системы рассчитана на возможности работы в рамках текущей инфраструктуры информационной системы. DocsVision рассчитан на работу с Microsoft SQL Server в качестве базового хранилища информации. В качестве Системы маршрутизации документов может использоваться как Microsoft Exchange Server, так и любая система электронной почты, поддерживающая протокол SMTP. Основной сервер приложений DocsVision работает под управлением Microsoft Internet Information Server.

В качестве клиентских приложений может использоваться как Internet Explorer, так и Microsoft Outlook. Приложения DocsVision могут быть встроены в Microsoft Office SharePoint Server с использованием технологии Digital Dashboard WEB Part.

Система DocsVision реализована в открытой архитектуре. Все функции приложений DocsVision доступны с помощью программных интерфейсов, обеспечивающих беспрецедентную гибкость использования компонентов платформы для создания конкретных решений. Имеются средства для встраивания отдельных функций системы в уже используемые в компаниях приложения. Система позволяет объединить в единый комплекс все приложения создаваемые на его базе.

Изначальная и полная WEB совместимость системы, в частности, ее инструментов управления, позволяет использовать приложения DocsVision из любой точки мировой сети, наличие средств off-line маршрутизации документов позволяют подключать к работе с системой пользователей, взаимодействующих с офисом компании только посредством электронной почты. [6]

# ЯЗЫК PHP

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом.

Говоря проще, PHP это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере.

Аббревиатура PHP означает “Hypertext Preprocessor (Препроцессор Гипертекста)". Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц. Подробнее о преимуществах PHP можно узнать [здесь](http://php.su/php/?opport).

Важным преимуществом языка PHP перед такими языками, как язык Perl и C заключается в возможности создания HTML документов с внедренными командами PHP.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

PHP прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

Язык PHP постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков web -программирования, по крайней мере, в ближайшее время.

## Возможности PHP

Возможности PHP очень большие. Главным образом, область применения PHP сфокусирована на написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять всё то, что выполняет любая другая программа [CGI](http://php.su/learnphp/cgi/). Например, обрабатывать данных форм, генерировать динамические страницы, отсылать и принимать [cookies](http://php.su/phphttp/?cookies). Но PHP способен выполнять и множество других задач.

Существуют три основных области, где используется PHP:

Создание скриптов для выполнения на стороне сервера. PHP наиболее широко используется именно таким образом. Все, что вам понадобится, это [парсер PHP](http://php.su/download/?php" \t "_blank) (в виде программы CGI или серверного модуля), [вебсервер](http://php.su/download/?apache" \t "_blank) и браузер. Чтобы Вы могли просматривать результаты выполнения PHP-скриптов в браузере, вам нужен работающий вебсервер и установленный PHP.

Создание скриптов для выполнения в [командной строке](http://php.su/prepare/?commandline). Вы можете создать PHP-скрипт, способный запускаться вне зависимости от вебсервера и браузера. Все, что вам потребуется - [парсер PHP](http://php.su/download/?php" \t "_blank). Такой способ использования PHP идеально подходит для скриптов, которые должны выполняться регулярно, например, с помощью cron (на платформах \*nix или [Linux](http://www.linux.org/" \t "_blank)) или с помощью планировщика задач (Task Scheduler) на платформах [Windows](http://windows.microsoft.com/" \t "_blank). Эти скрипты также могут быть использованы в задачах простой обработки текстов.

Создание приложений GUI, выполняющихся на стороне клиента. Возможно, PHP является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете PHP и хотели бы использовать некоторые его возможности в своих клиент-приложениях, вы можете использовать PHP-GTK для создания таких приложений. Подобным образом Вы можете создавать и кросс-платформенные приложения. PHP-GTK является расширением PHP и не поставляется вместе с дистрибутивом PHP. Если вы заинтересованы, посетите [сайт PHP-GTK](http://gtk.php.net/).

PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие, как HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS, и многих других. (Существует даже версия PHP для OS/2. Неизвестно, правда, насколько соответствующая нынешним реалиям). Также в PHP включена поддержка большинства современных вебсерверов, таких, как Apache, Microsoft Internet Information Server, Personal Web Server, серверов Netscape и iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd и многих других. Для большинства серверов PHP поставляется в качестве модуля, для других, поддерживающих стандарт CGI, PHP может функционировать в качестве процессора CGI.

Таким образом, выбирая PHP, вы получаете свободу выбора операционной системы и вебсервера. Кроме того, у вас появляется выбор между использованием процедурного или объектно-ориентированного программирования или же их сочетания.

PHP способен не только выдавать HTML. Возможности PHP включают формирование изображений, файлов PDF и даже роликов Flash (с использованием libswf и Ming), создаваемых "на лету". PHP также способен выдавать любые текстовые данные, такие, как XHTML и другие XML-файлы. PHP способен осуществлять автоматическую генерацию таких файлов и сохранять их в файловой системе вашего сервера, вместо того, чтобы отдавать клиенту, организуя, таким образом, кеш динамического содержания, расположенный на стороне сервера.

Одним из значительных преимуществ PHP является поддержка широкого круга [баз данных](http://php.su/phpdb/). Создание скрипта, использующего базы данных, очень просто. В настоящее время PHP поддерживает следующие базы данных:

|  |  |  |
| --- | --- | --- |
| Adabas D | Ingres | Oracle (OCI7 и OCI8) |
| dBase | InterBase | Ovrimos |
| Empress | FrontBase | PostgreSQL |
| FilePro (только чтение) | mSQL | Solid |
| Hyperwave | Direct MS-SQL | Sybase |
| IBM DB2 | MySQL | Velocis |
| Informix | ODBC | Unix dbm |

Также в PHP включена поддержка DBX для работы на абстрактном уровне, так что вы можете работать с любой базой данных, использующей DBX. Кроме того, PHP поддерживает ODBC (Open Database Connection standard), таким образом, вы можете работать с любой базой данных, поддерживающей этот всемирно признанный стандарт.

PHP также поддерживает "общение" с другими сервисами с использованием таких протоколов, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) и многих других. Кроме того, вы получаете возможность работать с сетевыми сокетами "напрямую". PHP поддерживает стандарт обмена сложными структурами данных WDDX. Обращая внимание на взаимодействие между различными языками, следует упомянуть о поддержке объектов [Java](http://java.sun.com/" \t "_blank) и возможности их использования в качестве объектов PHP. Для доступа к удаленным объектам вы можете использовать расширение CORBA.

PHP включает средства обработки текстовой информации, начиная с регулярных выражений Perl или POSIX Extended и заканчивая парсером документов XML. Для парсинга XML используются стандарты SAX и DOM. Для преобразования документов XML вы можете использовать расширение XSLT.

Последним по порядку, но не по значению, является поддержка многих других расширений, таких как функции поисковой машины [mnoGoSearch](http://www.mnogosearch.org/" \t "_blank), функции IRC Gateway, функции для работы со сжатыми файлами (gzip, bz2), функции календарных вычислений, функции перевода и многое другое.

## Преимущества PHP

Главным фактором языка [РНР](http://php.su/php/?php) является практичность. РНР должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер РНР обусловлен пятью важными характеристиками:

1. традиционностью;
2. простотой;
3. эффективностью;
4. безопасностью;
5. гибкостью.

Существует еще одна «характеристика», которая делает РНР особенно привлекательным: он распространяется бесплатно! Причем, с открытыми исходными кодами ( Open Source ).

Традиционность.

Язык РНР будет казаться знакомым программистам, работающим в разных областях. Многие конструкции языка позаимствованы из Си, Perl.

Код РНР очень похож на тот, который встречается в типичных программах на С или Pascal. Это заметно снижает начальные усилия при изучении РНР. PHP — язык, сочетающий достоинства Perl и Си и специально нацеленный на работу в Интернете, язык с универсальным (правда, за некоторыми оговорками) и ясным синтаксисом.

И хотя PHP является довольно молодым языком, он обрел такую популярность среди web-программистов, что на данный момент является чуть ли не самым популярным языком для создания web-приложений (скриптов).

Простота.

Сценарий РНР может состоять из 10 000 строк или из одной строки — все зависит от специфики вашей задачи. Вам не придется подгружать библиотеки, указывать специальные параметры компиляции или что-нибудь в этом роде. Механизм РНР просто начинает выполнять код после первой экранирующей последовательности (<?) и продолжает выполнение до того момента, когда он встретит парную экранирующую последовательность (?>). Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

PHP — язык, который может быть встроен непосредственно в html -код страниц, которые, в свою очередь будут корректно обрабатываться PHP -интерпретатором. Мы можем использовать PHP для написания CGI-сценариев и избавиться от множества неудобных операторов вывода текста. Мы можем привлекать PHP для формирования HTML-документов, избавившись от множества вызовов внешних сценариев.

Большое разнообразие функций PHP избавят вас от написания многострочных пользовательских функций на C или Pascal .

Эффективность.

Эффективность является исключительно важным фактором при программировании для многопользовательских сред, к числу которых относится и web .

Очень важное преимущество PHP заключается в его [«движке»](http://php.su/php/?phpengine). «Движок» PHP не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью.

По некоторым оценкам, большинство PHP-сценариев (особенно не очень больших размеров) обрабатываются быстрее аналогичных им программ, написанных на Perl. Однако, чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут работать значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных web-приложений. Подробно об устройстве и характеристиках «движка» PHP можно ознакомиться [здесь](http://php.su/php/?phpengine).

Безопасность.

РНР предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные [средства безопасности](http://php.su/security/), которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

1. Средства безопасности системного уровня

В РНР реализованы механизмы безопасности, находящиеся под управлением администраторов; при правильной настройке РНР это обеспечивает максимальную свободу действий и безопасность. РНР может работать в так называемом безопасном режиме (safe mode), который ограничивает возможности применения РНР пользователями по ряду важных показателей. Например, можно ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно влияет на быстродействие сервера). По аналогии с cgi-bin администратор также может устанавливать ограничения на каталоги, в которых пользователь может просматривать и исполнять сценарии РНР, а также использовать сценарии РНР для просмотра конфиденциальной информации на сервере (например, файла passwd).

2. Средства безопасности уровня приложения

В стандартный набор функций РНР входит ряд надежных механизмов шифрования. РНР также совместим с многими приложениями независимых фирм, что позволяет легко интегрировать его с защищенными технологиями электронной коммерции (e-commerce). Другое преимущество заключается в том, что исходный текст сценариев РНР нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация РНР на стороне сервера предотвращает похищение нетривиальных сценариев пользователями, знаний которых хватает хотя бы для выполнения команды View Source.

Гибкость.

Поскольку РНР является встраиваемым (embedded) языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя РНР обычно рекомендуется использовать в сочетании с HTML, он с таким же успехом интегрируется и в JavaScript, WML, XML и другие языки. Кроме того, хорошо структурированные приложения РНР легко расширяются по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем и с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии РНР полностью компилируются на стороне сервера. В сущности, сценарии РНР могут передаваться любым устройствам с браузерами, включая сотовые телефоны, электронные записные книжки, пейджеры и портативные компьютеры, не говоря уже о традиционных ПК. Программисты, занимающиеся вспомогательными утилитами, могут запускать РНР в [режиме командной строки](http://php.su/prepare/?commandline).

Поскольку РНР не содержит кода, ориентированного на конкретный web-сервер, пользователи не ограничиваются определенными серверами (возможно, незнакомыми для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold и Zeus — РНР работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, РНР в целом является платформенно-независимым языком и существует на таких платформах, как UNIX, Solaris, FreeBSD и Windows 95/98/NT/2000/XP/2003.

Наконец, средства РНР позволяют программисту работать с внешними компонентами, такими как Enterprise Java Beans или СОМ-объекты Win32. Благодаря этим новым возможностям РНР занимает достойное место среди современных технологий и обеспечивает масштабирование проектов до необходимых пределов.

Бесплатное распространение.

Стратегия Open Source, и распространение исходных текстов программ в массах, оказало несомненно благотворное влияние на многие проекты, в первую очередь — [Linux](http://www.linux.org/" \t "_blank), хотя и успех проекта [Apache](http://www.apache.org/" \t "_blank) сильно подкрепил позиции сторонников Open Source. Сказанное относится и к истории создания РНР, поскольку поддержка пользователей со всего мира оказалась очень важным фактором в развитии проекта РНР.

Принятие стратегии Open Source и бесплатное распространение исходных текстов РНР оказало неоценимую услугу пользователям. Вдобавок, отзывчивое сообщество пользователей РНР является своего рода «коллективной службой поддержки», и в популярных электронных конференциях можно найти ответы даже на самые сложные вопросы.

## История развития PHP

PHP прошёл долгий путь за последние несколько лет, становясь одним из наиболее популярных языков web-разработки.

Истоки PHP лежат в старом продукте, имевшем название PHP/FI. PHP/FI был создан Расмусом Лердорфом в 1995 году и представлял собой набор Perl-скриптов для ведения статистики посещений его резюме. Развитие web еще только начиналось, никаких специальных средств для решения этих задач не было, и к автору хлынул поток сообщений с вопросами. Лердорф начал бесплатно раздавать свой инструментарий, названный «Personal Homepages Tools» (PHP) - («Инструменты для персональных домашних страниц»). Очень скоро потребовалась большая функциональность и Расмус пишет новую, намного более обширную версию на C, работающую с базами данных и позволяющую пользователям разрабатывать простейшие web-приложения. Расмус Лердорф решил выложить исходный код PHP/FI на всеобщее обозрение, исправление ошибок и дополнение.

PHP/FI (Personal Home Page / Forms Interpreter - Персональная Домашняя страница / Интерпретатор Форм) включал в себя базовую функциональность сегодняшнего PHP. Он имел переменные в стиле Perl, автоматическую интерпретацию форм и возможность встраиваться в html-код. Собственно синтаксис языка имел много общего с Perl, хотя и был намного проще и ограниченнее.

В 1997 выходит PHP/FI 2.0. Вторая версия C-имплементации обозначила группу пользователей: несколько тысяч людей по всему миру, с примерно 50,000 доменами, что составляло около 1% всего числа доменов Интернета. Несмотря на то, что разработкой занималось уже несколько людей, PHP/FI 2.0 все еще оставался крупным проектом одного человека.

Официально PHP/FI 2.0 вышел только в ноябре 1997 года, после проведения большей части своей жизни в бета-версиях. Вскоре после выхода его заменили альфа-версии PHP 3.0.

PHP3

PHP 3.0 была первой версией, напоминающей PHP, каким мы знаем его сегодня. В 1997 году Энди Гутманс (Andi Gutmans) и Зив Сураски (Zeev Suraski) переписали код с начала: разработчики сочли PHP/FI 2.0 не пригодным для разработки приложения электронной коммерции, над которым они работали для проекта Университета. Для совместной работы над PHP 3.0 с помощью базы разработчиков PHP/FI 2.0 Энди, Расмус и Зив решили объединиться и объявить PHP 3.0 официальным преемником PHP/FI, разработка же PHP/FI была практически полностью прекращена.

Одной из сильнейших сторон PHP 3.0 была возможность расширения ядра. В последствии интерфейс написания расширений привлек к PHP множество сторонних разработчиков, работающих над своими модулями, что дало PHP возможность работать с огромным количеством баз данных, протоколов, поддерживать большое число API. Фактически, это и был главный ключ к успеху, но стоит добавить, что немаловажным шагом оказалась разработка нового, намного более мощного и полного синтаксиса с поддержкой [ООП](http://php.su/learnphp/phpoo/).

Абсолютно новый язык программирования получил новое имя. Разработчики отказались от дополнения о персональном использовании, которое имелось в аббревиатуре PHP/FI. Язык был назван просто 'PHP' -- аббревиатура, содержащая рекурсивный акроним: 'PHP: Hypertext Preprocessor' (PHP: Препроцессор Гипертекста).

Первая статья о PHP была опубликована в чешском варианте 'Computerworld' весной 1998 и освещала PHP 3.0. Как и в случае с книгами, эта была первая в серии статья из множества посвященных PHP и опубликованных в различных известных журналах.

К концу 1998, PHP использовался десятками тысяч пользователей. Сотни тысяч web-сайтов сообщали о том, что они работают с использованием языка. В то время PHP 3.0 был установлен приблизительно на 10% серверах Интернета!

PHP 3.0 был официально выпущен в июне 1998 года после 9 месяцев публичного тестирования.

PHP4

К зиме 1998 года, практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зив Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности базиса кода PHP. Расширения дали PHP 3.0 возможность успешно работать с набором баз данных и поддерживать большое количество различных API и протоколов, но PHP 3.0 не имел качественной поддержки модулей и приложения работали не эффективно.

Новый движок, названный 'Zend Engine' ([www.zend.com](http://www.zend.com/)) (от имен создателей: Zeev и Andi), успешно справлялся с поставленными задачами и впервые был представлен в середине 1999 года. PHP 4.0, основанный на этом движке и принесший с собой набор дополнительных функций, официально вышел в мае 2000 года, почти через два года после выхода своего предшественника PHP 3.0. В дополнение к улучшению производительности, PHP 4.0 имел еще несколько ключевых нововведений, таких как поддержка сессий, буферизация вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

Сегодня PHP используется сотнями тысяч разработчиков. Миллионы (!) сайтов сообщают о работе с PHP. Эти, а также многие другие факты, позволяют с уверенностью сказать, что PHP является одним из самых популярных языков web-программирования!

Группа разработчиков PHP состоит из множества людей, работающих над ядром и расширениями PHP, и смежными проектами, такими, как PEAR и документации языка.

PHP5

Недавно вышла новая, пятая версия PHP ([PHP5](http://php.su/php/?php5)). PHP5 использует новую версию «движка» Zend – Zend Engine 2.

В PHP5 объектная модель была значительно переработана. При этом было добавлено много новых возможностей, благодаря которым PHP5 получил некоторые черты таких объектно-ориентированных языков, как C++ и Java.

Изменения коснулись производительности встроенных стандартных функций PHP. По некоторым оценкам, скорость работы стандартных функций в PHP5 увеличилась на 40% и более.

Появились новые директивы файла настройки [php.ini](http://php.su/prepare/?phpini) .Претерпела изменения и поддержка потоков. Были добавлены новые функции, а также функции графической библиотеки GD.

Поддержка XML в версии PHP5 стала полной, поддерживаются новые расширения DOM и XML.

Что нового в PHP5 ?

Не так давно вышла новая пятая версия PHP - PHP5. Рассмотрим основные изменения.

Новая объектная модель Zend 2.0

Zend 1.0, а также более ранние версии "движка" PHP, нерационально использовали память, поскольку объекты обрабатывались как примитивные типы (переменные, списки), то есть посредством копирования. В новой объектной модели, которая используется в Zend 2.0 используется технология объектных дескрипторов, напоминающая аналогичную технологию в Java. Теперь объекты доступны только через дескрипторы, позволяя экономить память и получать полный контроль над созданием и дублированием объектов. О новых возможностях объектно-ориентированного программирования в PHP5 вы можете ознакомиться в [специальном разделе](http://php.su/learnphp/phpoo/?php5).

Улучшенная производительность

Zend Engine 2 использует новый менеджер памяти. Два главных его преимущества: улучшенная поддержка многопоточных сред (распределенные блоки не нуждаются во взаимных исключительных блокировках) и то, что после каждого запроса намного эффективнее освобождаются распределенные блоки памяти.

Изменение коснулись производительности встроенных функций PHP. По различным оценкам, увеличение производительности достигает 40%.

Новые директивы

В конфигурационном файле PHP - php.ini теперь доступны новые директивы:

session.hash\_function - выбор функции хеширования сессий (SHA1 или MD5);

session.hash\_bits\_per\_character - определяет, какое число бит в символе используется при хешировании сессий (от 4 до 6);

register\_long\_arrays - включает/отключает регистрацию длинных массивов;

mail.force\_extra\_parameters - дополнительные почтовые параметры;

zend.ze1\_compatibility\_mode - включение совместимости с Zend Engine 1 (PHP4).

Новые механизмы работы с потоками

Поддержка потоков претерпела множество изменений:

Более высокая производительность функций readfile(), fpassthru() и некоторых других стандартных функций для работы с потоками;

Новая функция stream\_socket\_client() - более мощный аналог сетевой функции fsockopen();

Новая функция stream\_socket\_accept() - принимает соединения клиента;

Новая функция stream\_socket\_get\_name() - получает локальное или удаленное имя сокета;

Новая функция stream\_copy\_to\_stream() - позволяет копировать поток в поток;

Новая функция, позволяющая читать информацию из потока до определенного символа или строки -stream\_get\_line();

Поддержка динамической загрузки OpenSSL;

Возможность работы функций opendir(), stat(), unlink() с FTP - соединениями.

Новы функции PHP5

В PHP5 включено достаточно большое количество новых функций:

[Функции для работы с массивами (Array):](http://php.su/functions/?cat=array)

[array\_combine()](http://php.su/functions/?array_combine) - Создает массив, используя один массив для ключей и другой - для его значений;

[array\_diff\_uassoc()](http://php.su/functions/?array_diff_uassoc) - Вычисляет различие массивов с дополнительной индексной проверкой;

[array\_udiff()](http://php.su/functions/?array_udiff) - Вычисляет различие массивов, используя функцию повторного вызова для сравнения данных;

[array\_udiff\_assoc()](http://php.su/functions/?array_udiff_assoc) - Вычисляет различие массивов с дополнительной индексной проверкой. Данные сравниваются с использованием функции повторного вызова;

[array\_udiff\_uassoc()](http://php.su/functions/?array_udiff_uassoc) - Вычисляет различие массивов с дополнительной индексной проверкой. Данные сравниваются с использованием функции повторного вызова. Индексная проверка также сделана функцией повторного вызова;

[array\_walk\_recursive()](http://php.su/functions/?array_walk_recursive) - Применяет пользовательскую функцию рекурсивно к каждому члену массива;

[array\_uintersect\_assoc()](http://php.su/functions/?array_uintersect_assoc) - Вычисляет пересечение массивов с дополнительной индексной проверкой. Данные сравниваются, используя функцию повторного вызова;

[array\_uintersect\_uassoc()](http://php.su/functions/?array_uintersect_uassoc) - Вычисляет пересечение массивов с дополнительной индексной проверкой. И данные, и индексы сравниваются, используя функции повторного вызова;

[array\_uintersect()](http://php.su/functions/?array_uintersect) - Вычисляет пересечение массивов. Данные сравниваются, используя функцию повторного вызова;

[Новые функции для работы с базой данных InterBase](http://php.su/functions/?cat=ibase):

[ibase\_affected\_rows()](http://php.su/functions/?ibase_affected_rows) - Возвращает номер строк, которые затронул предыдущий запрос;

[ibase\_backup()](http://php.su/functions/?ibase_backup) - Запускает задачу резервирования в сервисном менеджере;

[ibase\_commit\_ret()](http://php.su/functions/?ibase_commit_ret) - Производит транзакцию без ее закрытия;

[ibase\_db\_info()](http://php.su/functions/?ibase_db_info) - Запрос статистика базы данных;

[ibase\_drop\_db()](http://php.su/functions/?ibase_drop_db) - Удаляет базу данных;

[ibase\_errcode()](http://php.su/functions/?ibase_errcode) - Возвращает код ошибки;

[ibase\_free\_event\_handler()](http://php.su/functions/?ibase_free_event_handler) - Отменяет зарегистрированный обработчик события;

[ibase\_gen\_id()](http://php.su/functions/?ibase_gen_id) - Увеличивает значение генератора и возвращает его новое значение;

[ibase\_maintain\_db()](http://php.su/functions/?ibase_maintain_db) - Выполняет сервисную команду на сервере баз данных;

[ibase\_name\_result()](http://php.su/functions/?ibase_name_result) - Присваивает имя набору результатов;

[ibase\_num\_params()](http://php.su/functions/?ibase_num_params) - Возвращает число параметров в запросе;

[ibase\_param\_info()](http://php.su/functions/?ibase_param_info) - Возвращает информацию о параметре запроса;

[ibase\_restore()](http://php.su/functions/?ibase_restore) - Запускает задачу восстановления в сервисном менеджере;

[ibase\_rollback\_ret()](http://php.su/functions/?ibase_rollback_ret) - Производит "обратную транзакцию", сохраняя контекст транзакции;

[ibase\_server\_info()](http://php.su/functions/?ibase_server_info) - Запрашивает статистику базы данных;

[ibase\_service\_attach()](http://php.su/functions/?ibase_service_attach) - Осуществляет соединение с сервисным менеджером;

[ibase\_service\_detach()](http://php.su/functions/?ibase_service_detach) - Разрывает соединение с сервисным менеджером;

[ibase\_set\_event\_handler()](http://php.su/functions/?ibase_set_event_handler) - Регистрирует функцию повторного вызова, которая будет вызвана, когда события будут зарегистрированы;

[ibase\_wait\_event()](http://php.su/functions/?ibase_wait_event) - Ожидает события, которое будет зарегистрировано базой данных;

[Функции расширения iconv](http://php.su/functions/?cat=iconv):

[iconv\_mime\_decode()](http://php.su/functions/?iconv_mime_decode) - Декодирует поле заголовка MIME;

[iconv\_mime\_decode\_headers()](http://php.su/functions/?iconv_mime_decode_headers) - Декодирует множество поля заголовка MIME одновременно;

[iconv\_mime\_encode()](http://php.su/functions/?iconv_mime_encode) - Составляет поле заголовка MIME;

[iconv\_strlen()](http://php.su/functions/?iconv_strlen) - Возвращает число символов строки;

[iconv\_strpos()](http://php.su/functions/?iconv_strpos) - Находит позицию первого вхождения подстроки в строку;

[iconv\_strrpos()](http://php.su/functions/?iconv_strrpos) - Находит позицию последнего вхождения подстроки в строку;

[iconv\_substr()](http://php.su/functions/?iconv_substr) - Вырезает часть строки.

[Функции для работы с потоками (Streams)](http://php.su/functions/?cat=stream):

[stream\_copy\_to\_stream()](http://php.su/functions/?stream_copy_to_stream) - Копирование данных из одного потока, в другой;

[stream\_get\_line()](http://php.su/functions/?stream_get_line) - Получает строку из ресурса;

[stream\_socket\_accept()](http://php.su/functions/?stream_socket_accept) - Принимает соединение клиента на сокете, созданном[stream\_socket\_server()](http://php.su/functions/?stream_socket_server" \t "_blank);

[stream\_socket\_client()](http://php.su/functions/?stream_socket_client) - Отрывает соединение посредством Internet или Unix сокетов;

[stream\_socket\_get\_name()](http://php.su/functions/?stream_socket_get_name) - Возвращает имя локального или удаленного сокета;

[stream\_socket\_recvfrom()](http://php.su/functions/?stream_socket_recvfrom) - Получает данные из сокета, соединенного или нет;

[stream\_socket\_sendto()](http://php.su/functions/?stream_socket_sendto) - Посылает сообщение сокету, независимо, подключен он, или нет;

[stream\_socket\_server()](http://php.su/functions/?stream_socket_server) - Создает Internet или Unix сокет.

[Функции, используемые для работы с датами и временем](http://php.su/functions/?cat=datetime):

[idate()](http://php.su/functions/?idate) - Форматирует местное время/дату как целое число;

[date\_sunset()](http://php.su/functions/?date_sunset) - Время заката данного дня в зависимости от местоположения;

[date\_sunrise()](http://php.su/functions/?date_sunrise) - Время восхода данного дня в зависимости от местоположения;

[time\_nanosleep()](http://php.su/functions/?time_nanosleep) - Задержка в секундах и наносекундах.

[Функции для работы с строками (Strings](http://php.su/functions/?cat=strings)):

[str\_split()](http://php.su/functions/?str_split) - Преобразовывает строку в массив;

[strpbrk()](http://php.su/functions/?strpbrk) - Выполняет поиск строки с любым набором символов;

[substr\_compare()](http://php.su/functions/?substr_compare) - Выполняет регистро-независимое сравнение двух строк, начиная со смещения, и до длины строки;

Прочие новые функции:

[convert\_uudecode()](http://php.su/functions/?convert_uudecode) - Декодирует uuencoded - закодированную строку

[convert\_uuencode()](http://php.su/functions/?convert_uuencode) - Кодирует по типу uuencoded;

[curl\_copy\_handle()](http://php.su/functions/?curl_copy_handle) - Копирует дескриптор cURL;

[dba\_key\_split()](http://php.su/functions/?dba_key_split) - Разбивает ключ на строковое представление в виде массива;

[dbase\_get\_header\_info()](http://php.su/functions/?dbase_get_header_info) - Получает информацию о заголовке базы данных DBASE;

[dbx\_fetch\_row()](http://php.su/functions/?dbx_fetch_row) - Производит выборку строк из query\_result, который имеет флаг DBX\_RESULT\_UNBUFFERED;

[fbsql\_set\_password()](http://php.su/functions/?fbsql_set_password) - Изменяет пароль текущего пользователя базы данных fbsql;

[file\_put\_contents()](http://php.su/functions/?file_put_contents) - Записывает строку в файл;

[ftp\_alloc()](http://php.su/functions/?ftp_alloc) - Распределяет пространство для файла, который будет передан через FTP;

[get\_declared\_interfaces()](http://php.su/functions/?get_declared_interfaces) - Возвращает массив всех объявленных интерфейсов;

[get\_headers()](http://php.su/functions/?get_headers) - Получает все заголовки, посланные сервером в ответ на запрос HTTP;

[headers\_list()](http://php.su/functions/?headers_list) - Возвращает список посланных заголовков (или готовых к посылке);

[http\_build\_query()](http://php.su/functions/?http_build_query) - Генерирует URL-кодированную строку запроса;

[image\_type\_to\_extension()](http://php.su/functions/?image_type_to_extension) - Возвращает расширение файла графического типа, который был возвращен графическими функциями: [getimagesize()](http://php.su/functions/?getimagesize), [exif\_read\_data()](http://php.su/functions/?exif_read_data), [exif\_thumbnail()](http://php.su/functions/?exif_thumbnail), [exif\_imagetype()](http://php.su/functions/?exif_imagetype);

[imagefilter()](http://php.su/functions/?imagefilter) - Функция, позволяющая применять к графическим изображениям различные фильтры;

[imap\_getacl()](http://php.su/functions/?imap_getacl) - Возвращает ACL для данного почтового ящика;

[ldap\_sasl\_bind()](http://php.su/functions/?ldap_sasl_bind) - Связывает с каталогом LDAP, используя SASL;

[mb\_list\_encodings()](http://php.su/functions/?mb_list_encodings) - Возвращает массив всех поддерживаемых кодировок;

[pcntl\_getpriority()](http://php.su/functions/?pcntl_getpriority) - Возвращает приоритет любого процесса;

[pcntl\_wait()](http://php.su/functions/?pcntl_wait) - Ожидает возврата состояния текущей дочерней ветви, определенного системным вызовомwaitpid();

[pg\_version()](http://php.su/functions/?pg_version) - Возвращает массив с версиями клиента, протокола и сервера (когда доступно);

[php\_check\_syntax()](http://php.su/functions/?php_check_syntax) - Осуществляет проверку синтаксиса указанного PHP скрипта;

[php\_strip\_whitespace()](http://php.su/functions/?php_strip_whitespace) - Возвращает исходный код с разделенными комментариями и межстроковым пространством;

[proc\_nice()](http://php.su/functions/?proc_nice) - Изменяет приоритет текущего процесса;

[pspell\_config\_data\_dir()](http://php.su/functions/?pspell_config_data_dir) - Изменяет месторасположение файлов лингвистической информации;

[pspell\_config\_dict\_dir()](http://php.su/functions/?pspell_config_dict_dir) - Изменяет месторасположение основного списка слов;

[setrawcookie()](http://php.su/functions/?setrawcookie) - Посылает cookies без кодирования URL-а;

[snmp\_read\_mib()](http://php.su/functions/?snmp_read_mib) - Читает и производит парсинг MIB файла в активное MIB дерево;

[sqlite\_fetch\_column\_types()](http://php.su/functions/?sqlite_fetch_column_types) - Возвращает массив типов столбца конкретной таблицы;

Новые расширения

Расширение Tidy:

PHP 5 включает поддержку очень полезной библиотеки [Tidy](http://php.su/functions/?cat=tidy) (<http://tidy.sf.net/>). Она позволяет разработчикам PHP разбирать, диагностировать, чистить и восстанавливать документы HTML. Расширение Tidy поддерживает как функциональный, так и объектно-ориентированный интерфейс, и ее API использует механизм исключений РНР 5.

Расширение Perl:

Хотя и не включенное по умолчанию в РНР 5, расширение Perl позволяет вам вызывать Perl-скрипты, использовать объекты Perl и использовать другие функциональные возможности Perl прямо из кода PHP. Это новое расширение можно найти в репозитории PECL <http://pecl.php.net/package/perl/>.

Расширение SQLite:

Поддержка SQLite (<http://www.sqlite.org/>) изначально была введена в PHP 4.3.x. Это встроенная библиотека SQL, которая не требует SQL сервера и очень подходит для приложений, которые не требуют масштабируемых SQL-серверов, или если вы разворачиваете свое приложение у провайдера, который не предоставляет вам доступ к SQL-серверу. Вопреки названию, SQLite очень богата возможностями и поддерживает транзакции, вложенные выборки, представления (view) и большие DB-файлы. Здесь это упомянуто, как возможность PHP 5, потому что поддержка SQLite была введена довольно в поздних релизах PHP 4, и поскольку PHP 5 предоставляет новый объектно-ориентированный интерфейс и поддерживает итераторы.

Новое расширение MySQLi (усовершенствованная MySQL):

В PHP 5 для MySQL AB (<http://www.mysql.com/>) было написано новое расширение MySQL, которое позволяет вам полностью использовать преимущества новых функциональных возможностей в MySQL 4.1 и более новых версий. В противовес старому расширению MySQL новое дает вам возможность использовать оба интерфейса: функциональный и объектно-ориентированный. Так что теперь у вас есть выбор что предпочесть. Новые возможности, поддерживаемые этим расширением, включают контроль транзакций, поддержку репликаций, SSL и многое другое...

Улучшенная поддержка XML в PHP5

После изменений, касающихся самого языка, изменения в работе с XML в PHP 5 являются, вероятно, самыми существенными. Расширение функциональных возможностей XML в PHP 5 делают язык полностью равноправным другим, используемых в сети.

Основы:

Поддержка XML в PHP 4 базировалась на разнообразных библиотеках XML. Поддержка SAX осуществлялась старой библиотекой Expat, для использования XSLT необходима была библиотека Sablotron, для DOM использовалась более мощная libxml2 - библиотека проекта GNOME.

Использование разнообразных сторонних библиотек не делало PHP 4 лучше других языков, когда дело касалось поддержки XML. Сопровождение библиотек расширений было слабое, новые стандарты XML не всегда поддерживались, производительность была не столь хороша, как была возможна, взаимодействия между различными расширениями XML не существовало.

В PHP 5 все расширения XML были переписаны, чтобы использовать отличный набор инструментов, предоставляемый libxml2 (<http://www.xmlsoft.org/>). Это библиотека богатая возможностями, отлично сопровождаемая и эффективно реализовавшая стандарты XML, предоставляющая передовые возможности технологии XML в PHP.

Все вышеупомянутые расширения (SAX, DOM и XSLT) теперь используют libxml2, включая новые дополнительные расширения - SimpleXML и SOAP.

SAX:

Как говорилось выше, новая реализация SAX переведена от использования Expat к libxml2. Хотя новое расширение должно быть совместимо, могут быть некоторые тонкие различия. Разработчики, которые хотят продолжать работать с библиотекой Expat, могут это сделать, конфигурируя и собирая PHP соответствующим образом (не рекомендовано).

DOM:

Хотя поддержка DOM в PHP 4 была также основана на библиотеке libxml2, она изобиловала ошибками, грешила утечками памяти, и API во многих случаях не соответствовал рекомендациям W3C. Расширение DOM было полностью переписано для PHP 5. Мало того, что расширение было коренным образом переписано, теперь оно соответствует рекомендациям W3C. Например, названия функций теперь используют нотацию studlyCaps (присваивание имен со смешанным употреблением заглавных и строчных букв) как предписано по стандарту W3C, облегчая для вас применение прочитанного в документации W3C в PHP. Кроме того, расширение DOM теперь поддерживает три вида схем для валидации XML документов - DTD, XML Schema и RelaxNG.

В результате этих изменений код, написанный для PHP 4 с использованием DOM, не всегда будет выполняться в PHP 5. Однако, простая корректировка названий функций к новому стандарту в большинстве случаев решает проблему.

XSLT:

В PHP 4, было два расширения, которые поддерживали XSL-преобразования. Первое использовало Sablotron, а второе - поддержку XSLT в расширении DOM. В PHP 5 новое расширение XSL было основано на использовании libxml2. Так, в PHP 5 XSL-преобразование не принимает таблицу стилей XSLT в качестве параметра, но зависит от расширения DOM, чтобы загрузить ее. Таблица стилей может кэшироваться в памяти, и может быть применена ко многим документам для существенной экономии времени выполнения.

SimpleXML:

Вероятно, через год или два, оглянувшись назад, мы сможем сказать, что SimpleXML коренным образом изменил работу с XML-документами для PHP разработчиков. SimpleXML можно было бы действительно называть "XML для чайников". Вместо того, чтобы иметь дело с DOM или, что еще хуже, SAX, SimpleXML позволяет представить ваш XML-документ как родной объект PHP. Вы можете читать, писать или пробегаться по вашему XML-документу, с легкой непринужденностью получая доступ к элементам и атрибутам.

Рассмотрите следующий XML-документ:

<clients>   
<client>   
     <name>John Doe</name>   
     <account\_number>87234838</account\_number>   
</client>   
<client>   
     <name>Janet Smith</name>   
     <account\_number>72384329</account\_number>   
</client>   
</clients>

Следующий фрагмент кода печатает имя каждого клиента и номер его аккаунта:

<?php   
$clients = simplexml\_load\_file('clients.xml');   
foreach ($clients->client as $client) {   
     print "$client->name has account number $client->account\_number ";   
}   
?>

Как видим, SimpleXML действительно прост.

А в случае, если есть необходимость сделать что-то, что невозможно выполнить в рамках SimpleXML, вы можете преобразовать свой объект SimpleXML в дерево DOM, вызвав функцию dom\_import\_simplexml(), выполнить необходимое, и вернуться назад к SimpleXML, используя simplexml\_import\_dom(). Благодаря тому, что оба расширения реализованы в одной библиотеке XML, переключения между ними теперь реальны.

SOAP:

Официально поддержка SOAP в PHP 4 отсутствовала. Обычно, при необходимости использовать SOAP, приходилось использовать PEAR, но поскольку код был написан полностью на PHP, приложение не могло выполняться так же, как встроенное С-расширение. Другие доступные С-расширения так и не смогли достичь стабильной версии и широкого применения, и поэтому не были включены в PHP 5.

Поддержка SOAP в PHP 5 была полностью переписана как С-расширение и, хотя на текущий момент находится последней стадии в бета-тестирования, было решено включить в его стандартный дистрибутив, поскольку он практически полностью реализует стандарт SOAP.

Следующий код демонстрирует вызов функции SomeFunction(), определенной в WSDL-файле:

<?php   
$client = new SoapClient("some.wsdl");

$client->SomeFunction($a, $b, $c);   
?>

**[7]**

# ОСНОВНАЯ ИНФОРМАЦИЯ О БАЗАХ ДАННЫХ

База данных – представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов),систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) (Гражданский кодекс РФ, ст. 1260)

Существует множество других определений, отражающих скорее [субъективное](http://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D1%84%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F)) мнение тех или иных авторов о том, что означает база данных (БД) в их понимании, однако общепризнанная единая формулировка отсутствует. Наиболее часто используются следующие отличительные [признаки](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B7%D0%BD%D0%B0%D0%BA):

1. БД хранится и обрабатывается в [вычислительной системе](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80).  
   Таким образом, любые внекомпьютерные хранилища информации ([архивы](http://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D0%B2), [библиотеки](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0), [картотеки](http://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%82%D0%BE%D1%82%D0%B5%D0%BA%D0%B0) и т. п.) базами данных не являются.
2. Данные в БД логически структурированы ([систематизированы](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.  
   Структурированность подразумевает явное выделение составных частей ([элементов](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82)), связей между ними, а также [типизацию](http://ru.wikipedia.org/w/index.php?title=%D0%A2%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F&action=edit&redlink=1) элементов и связей, при которой с типом элемента (связи) соотносится определённая [семантика](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0) и допустимые операции.[[6]](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#cite_note-5)
3. БД включает [метаданные](http://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5), описывающие логическую структуру БД в формальном виде (в соответствии с некоторой [метамоделью](http://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)).  
   В соответствии с ГОСТ Р ИСО МЭК ТО 10032-2007, «постоянные данные в среде базы данных включают в себя [схему](http://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и базу данных. Схема включает в себя описания содержания, структуры и ограничений [целостности](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), используемые для создания и поддержки базы данных. База данных включает в себя набор постоянных данных, определенных с помощью схемы. Система управления данными использует определения данных в схеме для обеспечения доступа и управления доступом к данным в базе данных».

Из перечисленных признаков только первый является строгим, а другие допускает различные трактовки и различные степени оценки. Можно лишь установить некоторую степень соответствия требованиям к БД.

В такой ситуации не последнюю роль играет общепринятая практика. В соответствии с ней, например, не называют базами данных файловые архивы, [Интернет-порталы](http://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D0%BE%D1%80%D1%82%D0%B0%D0%BB&action=edit&redlink=1) или [электронные таблицы](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0), несмотря на то, что они в некоторой степени обладают признаками БД. Принято считать, что эта степень в большинстве случаев недостаточна (хотя могут быть исключения).

Многие специалисты указывают на распространённую ошибку, состоящую в некорректном использовании термина «база данных» вместо термина «[система управления базами данных](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)», и указывают на необходимость различения этих понятий.

Виды баз данных

Существует огромное количество разновидностей баз данных, отличающихся по различным [критериям](http://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9). Например, в «Энциклопедии технологий баз данных», по материалам которой написан данный раздел, определяются свыше 50 видов БД.

Основные классификации приведены ниже.

Классификация по [модели данных](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

1. [Иерархическая](http://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
2. [Сетевая](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
3. [Реляционная](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
4. [Объектная](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и [объектно-ориентированная](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
5. [Объектно-реляционная](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94)
6. [Функциональная](http://ru.wikipedia.org/w/index.php?title=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1).

Классификация по среде постоянного хранения

1. Во вторичной памяти, или традиционная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) conventional database): средой постоянного хранения является периферийная энергонезависимая память (вторичная память) — как правило[жёсткий диск](http://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA" \o "Жёсткий диск).  
   В оперативную память СУБД помещает лишь [кеш](http://ru.wikipedia.org/wiki/%D0%9A%D0%B5%D1%88" \o "Кеш) и данные для текущей обработки.
2. В оперативной памяти ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) in-memory database, memory-resident database, main memory database): все данные на стадии исполнения находятся в [оперативной памяти](http://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C).
3. В третичной памяти ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) tertiary database): средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило на основе[магнитных лент](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%B8%D0%BC%D0%B5%D1%80) или [оптических дисков](http://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA).  
   Во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кеш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры.

Классификация по содержимому

1. Географическая
2. Историческая
3. Научная
4. Мультимедийная.

Классификация по степени распределённости

1. Централизованная, или сосредоточенная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) centralized database): БД, полностью поддерживаемая на одном компьютере.
2. [Распределённая](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) distributed database): БД, составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.
3. Неоднородная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) heterogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами более одной СУБД
4. Однородная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) homogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами одной и той же СУБД.
5. Фрагментированная, или секционированная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) partitioned database): методом распределения данных является фрагментирование (партиционирование, [секционирование](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), вертикальное или горизонтальное.
6. Тиражированная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) replicated database): методом распределения данных является тиражирование ([репликация](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0))).

Другие виды БД:

1. [Пространственная](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [spatial database](http://meta.wikimedia.org/wiki/w:en:spatial_database" \o "m:w:en:spatial database)): БД, в которой поддерживаются пространственные свойства сущностей предметной области. Такие БД широко используются в[геоинформационных системах](http://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0).
2. Временная, или темпоральная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) temporal database): БД, в которой поддерживается какой-либо аспект времени, не считая времени, определяемого пользователем.
3. Пространственно-временная ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) spatial-temporal database) БД: БД, в которой одновременно поддерживается одно или более измерений в аспектах как пространства, так и времени.
4. [Циклическая](http://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) round-robin database): БД, объём хранимых данных которой не меняется со временем, поскольку в процессе сохранения данных одни и те же записи используются циклически.

Сверхбольшие базы данных

Сверхбольшая база данных ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Very Large Database, VLDB) — это база данных, которая занимает чрезвычайно большой объём на [устройстве физического хранения](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C). Термин подразумевает максимально возможные объёмы БД, которые определяются последними достижениями в технологиях физического хранения данных и в технологиях программного оперирования данными.

Количественное определение понятия «чрезвычайно большой объём» меняется во времени; в настоящее время считается, что это объём, измеряемый по меньшей мере [петабайтами](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%82%D0%B0%D0%B1%D0%B0%D0%B9%D1%82). Для сравнения, в 2005 г. самыми крупными в мире считались базы данных с объёмом хранилища порядка 100 [терабайт](http://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%80%D0%B0%D0%B1%D0%B0%D0%B9%D1%82).

Специалисты отмечают необходимость особых подходов к [проектированию](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) сверхбольших БД. Для их создания нередко выполняются специальные [проекты](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82) с целью поиска таких[системотехнических](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0) решений, которые позволили бы хоть как-то работать с такими большими объёмами данных. Как правило необходимы специальные решения для дисковой подсистемы, специальные версии [операционной среды](http://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0&action=edit&redlink=1) и специальные механизмы обращения [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) к данным.

Исследования в области хранения и обработки сверхбольших баз данных VLDB всегда находятся на острие теории и практики баз данных. В частности, с [1975 года](http://ru.wikipedia.org/wiki/1975_%D0%B3%D0%BE%D0%B4) проходит ежегодная конференция International Conference on Very Large Data Bases («Международная конференция по сверхбольшим базам данных»). Большинство исследований проводится под эгидой некоммерческой организации VLDB Endowment (Фонд целевого капитала «VLDB»), которая обеспечивает продвижение научных работ и обмен информацией в области сверхбольших БД и смежных областях. [8]

## MySQL

MySQL - это популярная система управления базами данных (СУБД), очень часто применяемая в сочетании с PHP.

База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми - от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному - как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL - это система управления реляционными базами данных. В реляционной базе данных данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

MySQL - это ПО с открытым кодом. Применять его и модифицировать может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями. Использование программного обеспечения MySQL регламентируется лицензией GPL (GNU General Public License), <http://www.gnu.org/licenses/>, в которой указано, что можно и чего нельзя делать с этим программным обеспечением в различных ситуациях.

Почему веб-программисты отдают предпочтение СУБД MySQL? MySQL является очень быстрым, надежным и легким в использовании. Если вам требуются именно эти качества, попробуйте поработать с данным сервером. MySQL обладает также рядом удобных возможностей, разработанных в тесном контакте с пользователями. Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

Технические возможности СУБД MySQL MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API). Мы также поставляем сервер MySQL в виде многопоточной библиотеки, которую можно подключить к пользовательскому приложению и получить компактный, более быстрый и легкий в управлении продукт. Доступно также большое количество программного обеспечения для MySQL, в большей части - бесплатного.

MySQL правильно произносится как ''Май Эс Кью Эль''.

Устройство MySQL

MySQL состоит из двух частей: серверной и клиентской.

Сервер MySQL постоянно работает на компьютере. Клиентские программы (например, скрипты PHP) посылают серверу MySQL SQL-запросы через механизм сокетов (то есть при помощи сетевых средств), сервер их обрабатывает и запоминает результат. То есть скрипт (клиент) указывает, какую информацио он хочет получить от сервера баз данных. Затем сервер баз данных посылает ответ (результат) клиенту (скрипту).  
Почему всегда передается не весь результат? Очень просто: дело в том, что размер результирующего набора данных может быть слишком большим, и на его передачу по сети уйдет чересчур много времени. Да и редко когда бывает нужно получать сразу весь вывод запроса (то есть все записи, удовлетворяющие выражению запроса). Например, нам может потребоваться лишь подсчитать, сколько записей удовлетворяет тому или иному условию, или же выбрать из данных только первые 10 записей. Механизм использования сокетов подразумевает технологию клиент-сервер, а это означает, что в системе должна быть запущена специальная программа — MySQL-сервер, которая принимает и обрабатывает запросы от программ. Так как вся работа происходит в действительности на одной машине, накладные расходы по работе с сетевыми средствами незначительны (установка и поддержание соединения с MySQL-сервером обходится довольно дешево).

Cтруктура MySQL трехуровневая: базы данных — таблицы — записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически - таблица представляет собой совокупность записей. А записи - это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы - в пределах базы данных, поля - в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя-идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей. [9]

MySQL является решением для малых и средних приложений. Входит в состав серверов [WAMP](http://ru.wikipedia.org/wiki/WAMP), [AppServ](http://ru.wikipedia.org/wiki/AppServ" \o "AppServ), [LAMP](http://ru.wikipedia.org/wiki/LAMP) и в портативные сборки серверов[Денвер](http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BD%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)), [XAMPP](http://ru.wikipedia.org/wiki/XAMPP). Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа [MyISAM](http://ru.wikipedia.org/wiki/MyISAM" \o "MyISAM), поддерживающие полнотекстовый поиск, так и таблицы [InnoDB](http://ru.wikipedia.org/wiki/InnoDB" \o "InnoDB), поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

26 февраля 2008 года [Sun Microsystems](http://ru.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) приобрела [MySQL AB](http://ru.wikipedia.org/wiki/MySQL_AB" \o "MySQL AB) за 1 миллиард долларов.[[3]](http://ru.wikipedia.org/wiki/MySQL#cite_note-2)

27 января 2010 года [Oracle Corporation](http://ru.wikipedia.org/wiki/Oracle_Corporation" \o "Oracle Corporation) приобрела [Sun Microsystems](http://ru.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) и включила MySQL в свою линейку СУБД.[[4]](http://ru.wikipedia.org/wiki/MySQL#cite_note-3)

Сообществом разработчиков MySQL созданы различные ответвления кода, такие как [Drizzle](http://ru.wikipedia.org/w/index.php?title=Drizzle&action=edit&redlink=1" \o "Drizzle (страница отсутствует)) ([англ.](http://en.wikipedia.org/wiki/Drizzle_(database_server))), [OurDelta](http://ru.wikipedia.org/w/index.php?title=OurDelta&action=edit&redlink=1" \o "OurDelta (страница отсутствует)), [Percona Server](http://ru.wikipedia.org/w/index.php?title=Percona_Server&action=edit&redlink=1" \o "Percona Server (страница отсутствует)), и [MariaDB](http://ru.wikipedia.org/wiki/MariaDB" \o "MariaDB). Все эти ответвления уже существовали на момент [поглощения](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B8%D1%8F%D0%BD%D0%B8%D1%8F_%D0%B8_%D0%BF%D0%BE%D0%B3%D0%BB%D0%BE%D1%89%D0%B5%D0%BD%D0%B8%D1%8F) компаний Sun и MySQL AB корпорацией Oracle.

О происхождении MySQL

MySQL возникла как попытка применить [mSQL](http://ru.wikipedia.org/wiki/MSQL" \o "MSQL) к собственным разработкам компании: таблицам, для которых использовались [ISAM](http://ru.wikipedia.org/wiki/ISAM) — подпрограммы низкого уровня. В результате был выработан новый [SQL](http://ru.wikipedia.org/wiki/SQL)-интерфейс, но [API](http://ru.wikipedia.org/wiki/API)-интерфейс остался в наследство от mSQL. Откуда происходит название «MySQL» — доподлинно неизвестно. Разработчики дают два варианта: либо потому, что практически все наработки компании начинались с префикса My, либо в честь девочки по имени My, дочери [Майкла Монти Видениуса](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BD%D0%B8%D1%83%D1%81,_%D0%9C%D0%B0%D0%B9%D0%BA%D0%BB), одного из разработчиков системы[[5]](http://ru.wikipedia.org/wiki/MySQL#cite_note-4)[[6]](http://ru.wikipedia.org/wiki/MySQL#cite_note-5).

Логотип MySQL в виде дельфина носит имя «[Sakila](http://ru.wikipedia.org/w/index.php?title=Sakila&action=edit&redlink=1" \o "Sakila (страница отсутствует))». Он был выбран из большого списка предложенных пользователями «имён дельфина». Имя «[Sakila](http://ru.wikipedia.org/w/index.php?title=Sakila&action=edit&redlink=1" \o "Sakila (страница отсутствует))» было отправлено [Open Source](http://ru.wikipedia.org/wiki/Open_Source" \o "Open Source)-разработчиком [Ambrose Twebaze](http://ru.wikipedia.org/w/index.php?title=Ambrose_Twebaze&action=edit&redlink=1" \o "Ambrose Twebaze (страница отсутствует)).

Платформы:

MySQL портирована на большое количество платформ: [AIX](http://ru.wikipedia.org/wiki/AIX), [BSDi](http://ru.wikipedia.org/wiki/BSDi" \o "BSDi), [FreeBSD](http://ru.wikipedia.org/wiki/FreeBSD), [HP-UX](http://ru.wikipedia.org/wiki/HP-UX), [Linux](http://ru.wikipedia.org/wiki/Linux), [Mac OS X](http://ru.wikipedia.org/wiki/Mac_OS_X), [NetBSD](http://ru.wikipedia.org/wiki/NetBSD" \o "NetBSD), [OpenBSD](http://ru.wikipedia.org/wiki/OpenBSD" \o "OpenBSD), [OS/2](http://ru.wikipedia.org/wiki/OS/2) Warp, [SGI](http://ru.wikipedia.org/wiki/SGI) [IRIX](http://ru.wikipedia.org/wiki/IRIX), [Solaris](http://ru.wikipedia.org/wiki/Solaris_(%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)), [SunOS](http://ru.wikipedia.org/wiki/SunOS), SCO OpenServer, SCO UnixWare, [Tru64](http://ru.wikipedia.org/wiki/Tru64" \o "Tru64), [Windows 95](http://ru.wikipedia.org/wiki/Windows_95), [Windows 98](http://ru.wikipedia.org/wiki/Windows_98), [Windows NT](http://ru.wikipedia.org/wiki/Windows_NT), [Windows 2000](http://ru.wikipedia.org/wiki/Windows_2000), [Windows XP](http://ru.wikipedia.org/wiki/Windows_XP), [Windows Server 2003](http://ru.wikipedia.org/wiki/Windows_Server_2003), WinCE, [Windows Vista](http://ru.wikipedia.org/wiki/Windows_Vista) и [Windows 7](http://ru.wikipedia.org/wiki/Windows_7). Существует также порт MySQL к [OpenVMS](http://ru.wikipedia.org/wiki/OpenVMS" \o "OpenVMS). Важно отметить, что на официальном сайте СУБД для свободной загрузки предоставляются не только исходные коды, но и откомпилированные и оптимизированные под конкретные операционные системы готовые исполняемые модули [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) MySQL.

Языки программирования:

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, PureBasic, Python, Ruby, Smalltalk, Компонентный Паскаль и Tcl библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.[10]

# ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

## Концепция экономического обоснования научно-технической разработки

Цель дипломного проекта – разработка системы для автоматизации обработки запросов заказов на швейном предприятии. Причина возникновения задачи по совершенствованию продукции и технологии ее производства продиктована современными требованиями к ведению бизнеса, технологии обработки заказов. Не для кого не секрет, что успех современной динамично развивающейся компании во многом зависит от технологий применяемых на данном предприятии. Сегодня в условиях жесткой конкуренции во всех сферах и отраслях производства, в том числе и в швейном, несомненным преимуществом является использование различных технических решений для оптимизации и автоматизации процесса производства, обработки заказов, улучшения производительности труда. Мы сфокусируем свое внимание на разработке системы для автоматизации обработки запросов заказов на швейном предприятии, что позволит нам решить следующие задачи: обеспечить оптимальное планирование рабочего дня работника; хранить все данные о заказах и их состоянии («принят», «сделан», «отдан», «отказ» и пр.) электронно; уменьшить время необходимое для подсчета выручки, зарплаты, кассы; интерактивность отображения незавершенных заказов, анализ прибыльных/убыточных дней, востребованных/невостребованных услуг и пр.

Цель технико-экономического обоснования - определение экономической целесообразности реализации проекта. Для решения поставленной цели необходимо изучить потребительские свойства продукта, связанного с совершенствованием продукта (технологии); провести тщательный анализ рынка и в соответствии с полученными данными составить подробный план маркетинга, затем описать процесс производства усовершенствованной продукции и осуществить оценку производственных ресурсов, необходимых для реорганизации производства продукции. Следующим этапом будет создание организационного плана проекта и оценка экономической эффективности разработки. Завершающей стадией ТЭО будет пункт «Выводы», в котором будут сформулированы основные результаты.

## Потребительские свойства продукта, связанного с совершенствованием продукта (технологии)

На данный момент все принятые заказы на швейном предприятии обрабатываются вручную. Это приводит к большим трудозатратам, неудобству в анализе принятых, сделанных и отданных заказов, сложности в отслеживании задолженностей клиентов. С вводом в эксплуатацию системы для автоматизации обработки запросов заказов на швейном предприятии мы решим часть данных проблем. Сравнительная оценка разрабатываемой продукции и продукции используемой в настоящее время представлена в табличной форме (табл. 1). Оценка приведена по пятибалльной шкале:

Таблица 1.

Технико-экономические параметры проектируемой продукции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Показатель | Балльная оценка | | Коэффициент значимости, µi |
| Традиционный документооборот, Ббаз | Новая система, Бi |
| 1 | Скорость прохождения документов | 5 | 9 | 0,1 |
| 2 | Вероятность ошибок | 6 | 9 | 0,2 |
| 3 | Трудозатраты на обработку документов | 3 | 8 | 0,3 |
| 4 | Возможность построения отчетов | 2 | 10 | 0,2 |
| 5 | Возможность контроля | 2 | 8 | 0,2 |

Уровень качества определяется по формуле: Ккач=∑ µi\*(Бi/ Ббаз) = 0,1\*9/5 + 0,2\*9/6 + 0,3\*8/3 + 0,2\*10/2 + 0,2\*8/2 = 3.08

Т.о. уровень качества новой системы превосходит уровень старой примерно в 3 раза

## Рынок и план маркетинга

Сегментирование рынка.

Все компании, занимающиеся пошивом и ремонтом одежды условно можно разделить на следующие категории:

1) Швейные предприятия, производящие текстильные изделия в крупных масштабах;

2) Ателье, специализирующиеся на индивидуальном пошиве;

3) Меховые ателье;

4) Ателье ремонта одежды.

Выбор целевого сегмента рынка

Сегодня динамично развиваются последние три вида швейных предприятий. Первый тип менее развит в связи с большой конкуренцией с китайскими и турецкими производителями за счет дешевой рабочей силы.

Поэтому наша система для автоматизации обработки запросов заказов на швейном предприятии в основном адресована именно ателье индивидуального пошива, ремонта одежда и меховых изделий. В результате маркетингового исследования, проведенного среди предприятий данного профиля была выведена потребность малых швейных предприятий в автоматизации процесса обработки заказов. Сейчас в подавляющем большинстве случаев используется традиционный ручной способ обработки заказов с использованием большого количества бумажной работы. Это негативно влияет на скорость выполнения заказов, ухудшает возможность их анализа, планирование рабочего дня сотрудника. В связи с этим для удовлетворения данных потребностей было принято решение разработать систему для автоматизации обработки запросов заказов на швейном предприятии и для проверки ее эффективности и работоспособности внедрить ее в компанию ООО «Сити-Сервис» - ателье по пошиву и ремонту одежды. В случае успешного внедрения планируется предлагать данную системы в другие профильные организации.

Выбор ценовой политики и оценка ожидаемых объемов продаж.

производится в целевых сегментах рынка и основаны на результатах маркетингового анализа. При выборе ценовой политики учитываются такие факторы, как рыночные (потребительские) свойства продукции, конкурентная структура рынка, маркетинговые цели и др. Цена будет устанавливаться намеренно заниженной (цена проникновения) с целью привлечения большего числа покупателей и захвата большей части рынка. Основные критерии по которым покупатель будет выбирать нашу продукции будут: цена, качество и сервис. Под ценой понимаются все ценовые условия, связанные с покупкой, в том числе цена отдельной поставки (start up price) – решено на начальной стадии продавать продукт по цене 10000 руб. (именно столько в среднем готовы платить малые предприятия – данные получены в результате маркетингового исследования); цена владения (ownership cost), включающая стоимость всех расходных материалов, комплектующих, необходимого окружения, связанного с использованием товара – для использования нашего системы клиенту клиенту понадобится оборудованное компьютерное место (компьютер, стол, стул). Качество в данном контексте определяется как возможность удовлетворения продукцией всех потребительских запросов, функциональную пригодность, безопасность, долговечность продукции, ее дизайн. Сервис (до- и послепродажное обслуживание) включает ознакомление потенциальных клиентов с товаром или услугой (education) – ознакомление будет производиться бесплатно, условия и сроки поставки – доставка будет осуществляться бесплатно, установка и наладка -бесплатно, обучение персонала – 5000 руб. курс для одного сотрудника , модернизация по запросу клиента – оговаривается индивидуально.

Сбыт новой продукции планируется в соответствии с ожидаемыми объемами продаж новой продукции и ее рыночными характеристиками.

Основные каналы сбыта:

1. Прямые продажи – торговый представитель будет ходить по ателье и предлагать нашу систему;
2. Интернет – сайт, реклама на досках объявлений, на тематических сайтах
3. Телефонные продажи

Продвижение товара планируется с целью стимулирования сбыта и включает анализ возможных мер воздействия на потенциальных покупателей для их привлечения к покупке новой продукции. В частности, следует выбрать эффективные методы рекламы новой продукции, предоставлять покупателям различные услуги, связанные с покупкой и использованием новой продукции (обучение персонала, предоставление скидок, гарантийное и послепродажное обслуживание клиентов и др.).

Эффективные методы сбыта:

1. Сбыт через торговых представителей, которые проводят деловые встречи с руководителями швейных предприятий, рекламируют наш продукт, описывают все его конкурентные преимущества;
2. Реклама в интернете;
3. Реклама в печатных тематических изданиях;
4. Рассылка писем с коммерческими предложениями по профильным организациям;
5. Предоставление различных скидок, бонусов в виде бесплатной доставке, установки, ознакомления продукции.

## Производство продукции

Подраздел “Производство продукции” предназначен для описания процесса производства усовершенствованной продукции и оценке производственных ресурсов, необходимых для реорганизации производства продукции. Результаты расчетов представим в табличной форме.

В данном подразделе:

* проведем описание изменений в производственном процессе или отдельных технологических операциях по изготовлению усовершенствованной продукции и оценим изменения в трудоемкости производства этой продукции с учетом планируемых объемов ее производства и сбыта. Сам процесс приема заказа изменится мало. Теперь все данные о принятом заказе (наименование, стоимость, планируемая дата выполнения) будут записаны в нашей разрабатываемой системе вместо бумажного варианта «Книги учета». Зато теперь в любой момент можно будет просмотреть историю и состояние принятых заказов, оптимально спланировать производственный процесс, провести анализ.
* проведем оценку стоимости технологического оборудования, которое высвобождается в процессе производства в связи с совершенствованием продукции (технологии).В связи с совершенствованием продукции (технологии) оборудование высвобождаться не будет. Оценим потребность в дополнительном оборудовании с учетом его производственной мощности, планируемых объемов производства – для использовании системы для автоматизации обработки запросов заказов на швейном предприятии понадобится оборудованное компьютерное место, включающее в себя стол, стул, компьютер.
* Рассчитаем размеры высвобождаемых производственных площадей в связи с совершенствованием продукции (технологии). Площади не высвобождаются, наоборот, нам понадобится дополнительная площадь для размещения компьютерного места.

Таблица 2.

Экономия производственных ресурсов в результате

совершенствования продукции (технологии)

(для текущих издержек – в год)

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование показателя экономии | Ед.изм. | Сумма экономии | |
| На ед. продукции | На объем производства |
| Экономия на текущих издержках, всего  в том числе: | р/заказ изд. | 6 | 6000 |
| Экономия потребности в материалах | р/заказ. | 0 | 0 |
| Экономия трудоемкости изготовления продукции | р/заказ. | 5 | 6000 |
| Прочие виды экономии текущих ресурсов | р/заказ. | 2 | 2400 |
| Экономия на потребности в основных средствах, всего  в том числе: | тыс.р. | 2 | 2000 |
| Сокращение потребности в технологическом  Оборудовании | тыс.р. | 0 | 0 |
| Сокращение потребности в производственных  Площадях | тыс.р. | 0 | 0 |
| Прочие виды экономии ресурсов | тыс.р. | 4 | 400 |

## Организационный план проекта

В данном разделе даем описание конкретных работ и организационных мероприятий, направленных на практическую реализацию проекта. Условно весь процесс можно разделить на три основных этапа –

1) составление, утверждение ТЗ, сбор и анализ данных по теме, изучение технологического процесса;

2) непосредственно реализация системы, обратная связь с пользователями;

3) Внедрение и сопровождение.

Более подробно весь процесс описан в табл.3.

Календарный план НИР

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Этапы и работы** | **Исполнитель** | **Трудоемкость, чел.-дн.** | **Численность, чел.** | **Длительность, дн.** | **Продолжительность работы (пятидневка)** | | | | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** |
| **1** | Составление и утверждение ТЗ | Рук.пр. | 2 | 1 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | Изучение ТЗ | Исп | 3 | 1 | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | Сбор и изучение документации по теме | Исп | 21 | 1 | 21 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | Анализ существующих технологий и библиотек для реализации автоматизированной системы документооборота | Исп | 7 | 1 | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | Изучение технологического процесса поставленной задачи | Исп | 12 | 1 | 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** | Разработка интерфейса системы, реализация прототипа системы | Исп | 30 | 1 | 30 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** | Доработка интерфейса, создание рабочей версии системы | Исп | 10 | 1 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** | Составление технической документации | Исп | 3 | 1 | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **№** | **Этапы и работы** | **Исполнитель** | **Трудоемкость, чел.-дн.** | **Численность, чел.** | **Длительность, дн.** | **Продолжительность работы (пятидневка)** | | | | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** |
| **9** | Анализ результатов | Исп | 4 | 1 | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Рукпр. | 4 | 1 | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** | Сдача документации | Исп | 1 | 1 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **11** | Внедрение | Исп | 1 | 1 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **12** | Сопровождение | Исп | 1 | ∞ | ∞ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Результатом работы является выполнение всех намеченных планов и достижение главной цели – разработка системы для автоматизации обработки запросов заказов на швейном предприятии.

Трудоемкость выполнения работы исполнителем составляет 93 чел.-дн, а руководителем проекта 6- чел.-дн. Общая продолжительность выполнения данной научно-исследовательской работы – 99 дней (14 недель).

## Экономическая эффективность разработки

В данном разделе дадим оценку интегрального экономического эффекта, получаемого от совершенствования продукции (технологии). При оценке экономического эффекта следует руководствоваться показателями, описанными в п. 3.1.7, с учетом особенностей их экономического содержания, определяемого спецификой объекта обоснования.

Применительно к обоснованию научно-технической разработки, направленной на совершенствование продукции (технологии) с целью снижения производственно-сбытовых издержек, в качестве инвестиционных затрат (I) выступают совокупные затраты на проведение научных исследований по совершенствованию продукции (технологии).

В качестве составляющих экономии выступают (см. табл. 12):

экономия на потребности в основных средствах;

экономия на текущих издержках.

В этом случае расчет интегрального экономического эффекта от внедрения ей результатов научно-технической разработки может быть рассчитан по следующей формуле:

 = 23901 руб.

где = 6000 руб.- экономия на текущих издержках; = 12000 руб.- единовременные затраты на проведение научных исследований и совершенствование продукции (технологии);  = 2000 руб. - экономия на потребности в основных средствах; R = 0,12– ставка дисконтирования, принятая для оценки анализируемого проекта; T = 10 – время реализации проекта, определяемое как количество интервалов инвестиционного периода, т.е. количество процентных периодов, по окончании которых производится начисление процентов.

Для оценки экономической эффективности может использоваться также следующее выражение, определяющее усредненную ежегодную экономию средств, связанных с внедрением результатов разработки:

 = 4230 руб.

где  - годовой экономический эффект от внедрения результатов совершенствования продукции (технологии).

В случае, если результаты разработки планируется использовать длительное время (), расчет годового экономического эффекта от внедрения результатов совершенствования продукции (технологии) может быть произведен по формуле

 = 4800 руб.

В данном разделе приведём также расчет основных показателей, характеризующих экономическую привлекательность проекта.

Для оценки экономической привлекательности дипломного проекта используем следующие показатели:

* рентабельность инвестиций;
* период возврата (срок окупаемости) инвестиций;
* чистая текущая стоимость проекта;
* внутренняя рентабельность проекта.

Показатель рентабельности инвестиций (ROI – Return Оn Investments)определяется как отношение среднегодовой прибыли к суммарным инвестиционным затратам в проект:

 = 6.9%

где  = 8321 руб. – чистая прибыль от проекта в году *t*; *T* = 10 – количество лет в инвестиционном периоде; *I = 12000 руб.* – величина инвестиционных затрат, связанных с реализацией проекта.

Рентабельность инвестиций может быть использована для сравнительной оценки эффективности проекта с альтернативными вариантами вложения капитала. Например, если рентабельность инвестиций, выраженная в процентах, превышает среднюю норму доходности по депозитным вкладам в коммерческие банки, то при сопоставимом уровне риска вложений можно утверждать, что вложение средств в проект является более привлекательным, чем хранение этих средств на депозитных счетах.

Период возврата (срок окупаемости) инвестиций () определяет промежуток времени от момента первоначального вложения капитала в проект, до момента, когда нарастающий итог суммарного чистого дохода становится равным нулю. Для определения периода возврата можно воспользоваться данными прогноза движения денежных средств и установить инвестиционный интервал, после которого показатель, определяемый как нарастающий итог ЧДП, становится положительной величиной. Этот интервал и определяет срок окупаемости инвестиций. Очевидно, что чем меньше период возврата инвестиций, тем проект является более экономически привлекательным.

Ток = 1.44

Показатель внутренней рентабельности проекта (*IRR* ‑ Internal Rate of Return) определяет такую ставку дисконта, при которой дисконтированная стоимость поступлений денежных средств по проекту равна дисконтированной стоимости платежей:

,

*IRR = 23%*

где *IRR* – искомая ставка внутренней рентабельности проекта.

Проект считается экономически выгодным, если внутренняя рентабельность превышает минимальный уровень рентабельности, установленный для данного проекта. Наш проект экономически выгоден. Кроме того, этот показатель определяет максимально допустимую ставку ссудного процента (стоимости заемного капитала), при которой кредитование проекта осуществляется безубыточно, т. е. без использования для выплат за кредит части прибыли, полученной на собственный инвестированный капитал. [11]

## Выводы

Получены следующие финансовые показатели проекта (инвестиционный период – 10 лет): себестоимость проекта – 8200 руб.; рентабельность инвестиций – 6.9%; чистая текущая стоимость проекта – 23901 руб. Коэффициент качества Ккач = 3,08. Полученные показатели дают достаточные основания для принятия решения о финансировании и реализации проекта. Разработанный проект даёт возможность не только получить доход от его продаж, но и существенно сэкономить время, трудозатраты и повысить надёжность обработки запросов заказов на швейном предприятии.

# [ОХРАНА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ](#_Toc157806229)

В этом разделе рассматриваются особенности программ для ЭВМ как объектов правовой охраны, а также вопросы защиты прав на них в рамках действующего законодательства. Раздел посвящен практическим вопросам, связанным с официальной регистрацией программ для ЭВМ в Роспатенте и особенностям их коммерческой реализации.

## Программа для ЭВМ как объект правовой охраны

Законодательно регулируются правоотношения, которые возникают в результате создания РИД. В процессе дипломного проекта мною, Подоплеловым Андреем, разработана программа для ЭВМ “ Система для автоматизации обработки запросов заказов на швейном предприятии”, которая относится к РИД.

Под программой для ЭВМ понимается "... представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определенного результата". Кроме того, в понятие программы для ЭВМ входят "...подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения".

С точки зрения программистов и пользователей программа для ЭВМ представляет собой детализацию алгоритма решения какой-либо задачи и выражена в форме определенной последовательности предписаний, обеспечивающих выполнение компьютером преобразования исходных данных в искомый результат.

Можно выделить следующие объективные формы представления программы для ЭВМ:

исходная программа (или исходный текст) - последовательность предписаний на алгоритмическом (понятном человеку) языке высокого уровня, предназначенных для автоматизированного перевода этих предписаний в последовательность команд в объектном коде;

рабочая программа (или объектный код) - последовательность машинных команд, т. е. команд, представленных на языке, понятном ЭВМ;

программа, временно введенная в память ЭВМ, - совокупность физических состояний элементов памяти запоминающего устройства ЭВМ (ОЗУ), сохраняющихся до прекращения подачи электропитания к ЭВМ;

программа, постоянно хранимая в памяти ЭВМ, - представленная на языке машины команда (или серия команд), выполненная в виде физических особенностей участка интегральной схемы, сохраняющихся независимо от подачи электропитания.

Исходная и рабочая программы, как правило, представляются в виде записи на том или ином языке, выполненной на бумаге или машиночитаемом носителе данных: магнитном или оптическом диске, магнитной ленте и т. п.

Предоставляемая законодательством правовая охрана распространяется "... на все виды программ для ЭВМ (в том числе на операционные системы и программные комплексы), которые могут быть выражены на любом языке и в любой форме, включая исходный текст и объектный код …". Так как преобразование исходного текста программы для ЭВМ в объектный (машинный) код с помощью специальных программ-трансляторов не меняет сущности данной программы как произведения, то если охраняется исходный текст программы, значит, охране подлежит и соответствующий ей объектный код. Обратное тоже справедливо.

Правовая охрана программ для ЭВМ распространяется только в отношении формы их выражения и «… не распространяется на идеи, концепции, принципы, методы, процессы, системы, способы, решения технических, организационных или иных задач, открытия, факты, языки программирования».

## 

## Официальная регистрация программ для ЭВМ

## 

В ГК РФ закреплено право автора или иного правообладателя на государственную регистрацию программы для ЭВМ. Исключение составляют программы для ЭВМ, в которых содержатся сведения, составляющие государственную тайну.

Процедура официальной регистрации программ для ЭВМ включает подачу заявки в федеральный орган исполнительной власти по интеллектуальной собственности (Роспатент), проверку поданных документов и собственно регистрацию. [12]

Заявка должна содержать следующие документы:

1. заявление о государственной регистрации;
2. депонируемые материалы, идентифицирующие программу для ЭВМ, включая реферат;
3. документ, подтверждающий уплату государственной пошлины в установленном размере или основание для освобождения от уплаты государственной пошлины или уменьшения его размера.

Ниже представлен заполненный двухсторонний бланк заявления на официальную регистрацию программы для ЭВМ, созданной в рамках данного дипломного проекта. Также представлены депонируемые материалы, идентифицирующие программу для ЭВМ, реферат на регистрируемую программу. Для заполнения приводимых в этом разделе документов программа была названа «Система для автоматизации обработки запросов заказов на швейном предприятии».

.

**ПРОГРАММА ДЛЯ ЭВМ**

**Система для автоматизации обработки запросов заказов на швейном предприятии**

Фрагменты исходного текста программы

Всего 12 листов

Правообладатель: Подоплелов Андрей Владимирович

Автор: Подоплелов Андрей Владимирович

(ф., и., о.)

(С) Подоплелов А.В., 2012

Санкт-Петербург

2012

**Р Е Ф Е Р А Т**

Автор: Подоплелов Андрей Владимирович

Правообладатель: Подоплелов Андрей Владимирович

Программа для ЭВМ: Система для автоматизации обработки запросов заказов на

швейном предприятии

Аннотация: Программный комплекс предназначен для автоматизации

обработки запросов заказов на швейном предприятии для

упрощения документооборота в компании. Программный

комплекс состоит из базы данных содержащей сведения о

принятых заказах и их состоянии и трех скриптов,

совмещающего в себе функции интерфейсного модуля и

управляющей программы.

Тип ЭВМ: IBM PC/AT и совместимые с ней

ОС: Windows 95/98/МЕ/NT/XP/Vista/7

Язык программирования: PHP 5.3.1

Объем: 14 Кбайт

Санкт-Петербург

2012

**СОСТАВ ПРОГРАММНОГО КОМПЛЕКСА**

Index.php – модуль оформления заказов (исходный текст)

Orders.php – модуль отображения незавершенных заказов (исходный текст)

Search.php – модуль поиска заказов (исходный текст)

Data.sql – база данных, содержащая сведения о принятых заказах.

## 6.3. Коммерческая реализация программ для ЭВМ

Коммерческая реализация программного продукта - программы для ЭВМ связана с понятием использования программы для ЭВМ третьими лицами и осуществляется на основании лицензионного договора с правообладателем.

Одним из типов лицензионного договора на программу для ЭВМ или базу данных является традиционный двухсторонний договор правообладателя – лицензиара, с покупателем (пользователем) - лицензиатом, в котором определяется способы, сроки, территория использования программы или базы данных. Такие договоры составляются, как правило, при:

▫ единичных продажах программного продукта, предназначенного для

решения достаточно узких прикладных задач;

▫ при продажах программного продукта, требующего регулярного

обновления и дополнения;

▫ передаче прав на тиражирование и распространение программного

продукта. [12]

Ниже приведён лицензионный договор, на основании которого я, Подоплелов Андрей, могу продать программный продукт “ Система для автоматизации обработки запросов заказов на швейном предприятии ” СПбГЭТУ “ЛЭТИ”.

**ЛИЦЕНЗИОННЫЙ ДОГОВОР**

**НА ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ДЛЯ ЭВМ**

**Стороны в Договоре:**

Гражданин **Подоплелов Андрей Владимирович, проживающий по адресу: 190000, г. Санкт-Петербург, ул. Лени Голикова, д. 47, кв. 3,** именуемый в дальнейшем "ЛИЦЕНЗИАР", с одной стороны, и

**Государственное образовательное учреждение высшего профессионального образования "Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И.Ульянова (Ленина)",** именуемый в дальнейшем "ЛИЦЕНЗИАТ", в лице в лице первого проректора по научно-образовательной деятельности КУТУЗОВА В. М., действующего на основании Устава, с другой стороны,

**принимая во внимание:**

1. что Лицензиар является автором и правообладателем программы для ЭВМ **"** **Система для автоматизации обработки запросов заказов на швейном предприятии ";**
2. Лицензиат желает получить на условиях настоящего Договора лицензию на использование упомянутой программы для ЭВМ с целью проведения научных исследований в области медицинского приборостроения;
3. Лицензиар готов предоставить Лицензиату такую лицензию,

договорились о следующем.

**1. Термины и их определения**

1. "ПРОГРАММА ДЛЯ ЭВМ (ПрЭВМ)" - программное обеспечение **"Система для автоматизации обработки запросов заказов на швейном предприятии**
2. "ДОКУМЕНТАЦИЯ" - комплект документов, передаваемых Лицензиаром Лицензиату, включающий руководство пользователя по применению и обслуживанию программы для ЭВМ.
3. "ПРОИЗВОДСТВЕННАЯ ПЛОЩАДКА" - **научные лаборатории и кафедры** Лицензиата.
4. "РАБОЧЕЕ МЕСТО" - конкретная ЭВМ, на которой используется Программа для ЭВМ.

**2. Предмет Договора**

1. Лицензиар предоставляет Лицензиату на срок действия настоящего Договора и за вознаграждение, уплачиваемое Лицензиатом, неисключительную лицензию на использование ПрЭВМ. При этом Лицензиату предоставляется право на установку ПрЭВМ не более чем на **10 (десяти**) Рабочих местах.
2. Лицензиар передает Лицензиату Документацию к ПрЭВМ.
3. Предоставленное Лицензиату в рамках настоящего Договора право ограничено Производственной площадкой.
4. Лицензиар осуществляет авторский контроль за соблюдением объемов использования ПрЭВМ по настоящему Договору, при этом Лицензиат обеспечивает возможность такого контроля.
5. Лицензиар сохраняет за собой право самому использовать ПрЭВМ и предоставлять неисключительные лицензии на право ее использования третьим лицам.

**3. Обеспечение Договора**

1. Лицензиар передает Лицензиату ПрЭВМ в объеме и виде, достаточном для ее использования, и Документацию в течение 15 (пятнадцати) дней со дня подписания настоящего Договора. ПрЭВМ передается Лицензиату в виде в количестве 5 (пяти) штук, содержащих ПрЭВМ. По факту передачи ПрЭВМ и Документации составляется акт сдачи-приемки с перечнем переданных материалов, подписываемый обеими Сторонами.
2. Если Лицензиат установит неполноту или неправильность полученных ПрЭВМ или Документации, то Лицензиар в течение 15 (пятнадцати) дней после сообщения ему об этом Лицензиатом обязан передать недостающие материалы или устранить недостатки ранее переданных ПрЭВМ и Документации.
3. Для оказания помощи в освоении ПрЭВМ Лицензиар по просьбе Лицензиата оказывает консультации пользователям ПрЭВМ.
4. Для целей использования ПрЭВМ в объеме, предусмотренном п. 2.1 настоящего Договора, Лицензиат может изготавливать в необходимом ему количестве копии ПрЭВМ и копии Документации.

**4. Усовершенствования**

1. Лицензиар обязуется незамедлительно информировать Лицензиата о всех произведенных им усовершенствованиях ПрЭВМ и, при желании Лицензиата, передать ему в согласованные сроки новые варианты ПрЭВМ. В отношении новых вариантов ПрЭВМ, переданных Лицензиаром Лицензиату, распространяются все условия настоящего Договора.
2. Лицензиат обязуется предоставлять Лицензиару информацию об использовании ПрЭВМ, которая могла бы быть полезной для усовершенствования ПрЭВМ.

**5. Платежи**

1. За предоставление прав, предусмотренных настоящим Договором, Лицензиат выплачивает Лицензиару единовременное вознаграждение в размере 20000(двадцать тысяч) рублей.
2. Вознаграждение, предусмотренное п. 5.1 настоящего Договора, выплачивается Лицензиатом в течение 30 (тридцати) дней, следующих после подписания акта приемки-сдачи.

**6. Реклама**

1. Лицензиат обязуется при опубликовании результатов исследований, полученных с использованием ПрЭВМ, сообщать в рекламных целях, что исследования производились с использованием ПрЭВМ Лицензиара с указанием авторского права Лицензиара.

**7. Защита передаваемых прав**

1. Лицензиат обязуется не вносить самовольно каких-либо изменений в ПрЭВМ и Документацию и не дополнять их какими-либо комментариями. Подобные изменения или дополнения возможны только с согласия Лицензиара.
2. Лицензиат обязуется предпринимать все необходимые меры для предотвращения несанкционированного копирования ПрЭВМ и Документации третьими лицами, а также несанкционированной передачи ПрЭВМ и Документации работниками Лицензиата третьим лицам.
3. Если Лицензиату станет известно о противоправном использовании ПрЭВМ третьими лицами, то он незамедлительно сообщит об этом Лицензиару.

**8. Ответственность Сторон и разрешение споров**

1. За невыполнение или ненадлежащее выполнение обязательств по настоящему Договору Стороны несут имущественную ответственность в соответствии с действующим законодательством.
2. Стороны освобождаются от ответственности за неисполнение или ненадлежащее исполнение обязательств, принятых по настоящему Договору, если неисполнение явилось следствием обстоятельств непреодолимой силы (форс-мажор).
3. Сторона, нарушившая свои обязательства по настоящему Договору, освобождается от ответственности за неисполнение или ненадлежащее исполнение этих обязательств, если это нарушение было вызвано причинами, за которые отвечает другая Сторона.
4. В случае возникновения споров между Лицензиаром и Лицензиатом по вопросам, предусмотренным настоящим Договором, Стороны примут все меры к разрешению их путем переговоров между собой. В случае невозможности разрешения указанных споров путем переговоров они будут разрешаться в порядке, предусмотренном действующим законодательством.

**9. Срок действия Договора и условия его расторжения**

1. Настоящий Договор заключен на срок 2 года и вступает в силу с даты его подписания обеими Сторонами.
2. По истечении срока действия настоящего Договора Лицензиат вправе использовать ПрЭВМ, включая усовершенствованные варианты, на Производственной площадке на любом количестве Рабочих мест. При этом обязательства Лицензиата, предусмотренные пп. 7.1 и 7.2 настоящего Договора, сохраняются бессрочно.
3. Действие настоящего Договора по обоюдному согласию Сторон может быть досрочно прекращено, но не ранее чем через три месяца после предложения об этом одной из Сторон. При этом Лицензиат не освобождается от обязательств по платежам, возникшим до расторжения настоящего Договора.
4. Настоящий Договор может быть досрочно расторгнут в одностороннем порядке со стороны Лицензиара из-за невыполнения Лицензиатом своих обязательств по пп. 7.1 или 7.2. В этом случае Лицензиат лишается права дальнейшего использования ПрЭВМ в любой форме и обязан вернуть ее Лицензиару.
5. Если Лицензиат откажется от дальнейшего использования ПрЭВМ, то он уничтожит все имеющиеся у него копии ПрЭВМ.

**10. Заключительные положения**

1. Все изменения и дополнения к настоящему Договору действительны только в тех случаях, если они совершены в письменной форме и подписаны обеими Сторонами.
2. Стороны не имеют права передавать свои права и обязательства по настоящему Договору третьим лицам без письменного согласия на то другой Стороны.
3. Во всем остальном, что не предусмотрено условиями настоящего Договора, будут применяться нормы законодательства Российской Федерации.

**11. Адреса Сторон**

1. ЛИЦЕНЗИАР: **Подоплелов Андрей Владимирович, проживающий по адресу: 190000, г. Санкт-Петербург, ул. Лени Голикова, д. 47, кв. 3**
2. ЛИЦЕНЗИАТ: **СПбГЭТУ, адрес: 197376, Санкт-Петербург, ул. Проф. Попова, д. 5.**

Настоящий Договор составлен в двух экземплярах для каждой из Сторон и подписан "\_\_\_"\_\_\_\_\_\_\_\_\_\_ 200\_\_ г. в г. Санкт-Петербурге.

|  |  |
| --- | --- |
| **ЛИЦЕНЗИАР:** | **От ЛИЦЕНЗИАТА:** |
| \_\_\_\_\_\_\_\_\_\_\_ А.В. Подоплелов | Первый проректор по научно-образовательной деятельности СПбГЭТУ  \_\_\_\_\_\_\_\_\_\_\_\_ В.М. КУТУЗОВ |

# ЗАКЛЮЧЕНИЕ

Причина возникновения задачи по совершенствованию продукции и технологии ее производства продиктована современными требованиями к ведению бизнеса, технологии обработки заказов. Не для кого не секрет, что успех современной динамично развивающейся компании во многом зависит от технологий применяемых на данном предприятии. Сегодня в условиях жесткой конкуренции во всех сферах и отраслях производства, в том числе и в швейном, несомненным преимуществом является использование различных технических решений для оптимизации и автоматизации процесса производства, обработки заказов, улучшения производительности труда. Мы сфокусировали свое внимание на разработке системы для автоматизации обработки запросов заказов на швейном предприятии, что позволило нам решить следующие задачи: обеспечить оптимальное планирование рабочего дня работника; хранить все данные о заказах и их состоянии («принят», «сделан», «отдан» и пр.) электронно; уменьшить время необходимое для подсчета выручки, зарплаты, кассы; интерактивность отображения незавершенных заказов, анализ прибыльных/убыточных дней, востребованных/невостребованных услуг и пр.

Также была оформлена пояснительная записка, включающая в себя следующие разделы:

1. Введение
2. Обзор существующих систем автоматизации документооборота
3. Язык PHP
4. Основная информация о базах данных
5. Технико-экономическое обоснование проекта
6. Охрана интеллектуальной собственности

Заключение

Список используемой литературы

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. http://ru.wikipedia.org/wiki/Система\_автоматизации\_документооборота
2. http://v8.1c.ru/overview/
3. http://v8.1c.ru/small.biz/
4. http://www.documetr.ru/anot.html
5. http://www.documetr.ru/uploads/Documetr\_pricelist\_20110501.pdf
6. http://www.docsvision.com/catalog/produkti/dv\_24.html
7. http://php.su/php/
8. Кузнецов С. Д. Основы баз данных — 2-е изд. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с.
9. http://php.su/mysql/?info
10. Стив Суэринг, Тим Конверс, Джойс Парк PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible — М.: «Диалектика», 2010. — 912 с
11. Васильев А. В. Методические указания по технико-экономическому обоснованию дипломных проектов (работ):Метод. указания - СПбГЭТУ «ЛЭТИ», - 2002. - 32с.
12. Правовая охрана и коммерческая реализация программ для ЭВМ и баз данных: Методические указания по дисциплине "Интеллектуальная собственность"/ Ю. И. Буч, И. С. Терентьева. - СПбГЭТУ. СПб. - 2008. - 62 с.

# ПРИЛОЖЕНИЕ 1

## Исходный текст модуля Index.php

<?php

$mysql\_host = "localhost";

$my\_database = "data";

$mysql\_user = "root";

echo "<h5 align='center'>Система для автоматизации обработки запросов заказов на швейном предприятии v.1.0</h5>";

echo "<h3 align='center'>Оформление заказов</h3>";

echo "<a href = 'index.php'> Оформление заказов</a> ";

echo "<a href = 'orders.php'> Незавершенные заказы</a> ";

echo "<a href = 'search.php'> Поиск по заказам</a><br><br>";

$cur\_day = date("Y.m.d");

echo "<b>Текущее число: ".$cur\_day."<br><br>";

echo "<b>Заказ выполнен:

<form action='' method='POST'>

<table border='1' width='20%' cellpadding='0'>

<tr>

<td> <p align='center'>№</td>

<td> <p align='center'>Дата фактическая</td>

</tr>

<tr>

<td> <p align='center'><input type='text' name='done\_number'></td>

<td> <p align='center'><input type='text' name='done\_date\_fact'></td>

</tr>

</table>

<input type='submit' value='Записать'>

</form>

<br>";

echo "<b>Оформить заказ:

<form action='' method='POST'>

<table border='1' width='100%' cellpadding='0'>

<tr>

<td> <p align='center'>№</td>

<td> <p align='center'>Описание</td>

<td> <p align='center'>Цена</td>

<td> <p align='center'>Клиент</td>

<td> <p align='center'>Дата приема</td>

<td> <p align='center'>Дата по договору</td>

<td> <p align='center'>Дата фактическая</td>

<td> <p align='center'>Мастер</td>

</tr>

<tr>

<td> <p align='center'><input type='text' name='number'></td>

<td> <p align='center'><input type='text' name='description'></td>

<td> <p align='center'><input type='text' name='price'></td>

<td> <p align='center'><input type='text' name='client'></td>

<td> <p align='center'><input type='text' name='date\_pr' value = '$cur\_day'></td>

<td> <p align='center'><input type='text' name='date\_dog'></td>

<td> <p align='center'><input type='text' name='date\_fact'></td>

<td> <p align='center'><input type='text' name='master'></td>

</tr>

</table>

<input type='submit' value='Оформить'>

</form><br>";

// Connecting, selecting database

$link = mysql\_connect($mysql\_host, $mysql\_user)

or die('Could not connect: ' . mysql\_error());

//echo 'Connected successfully<br>';

//$query = "CREATE DATABASE `my\_database`";

//$sql = mysql\_query ($query) or die('Could not create database: ' . mysql\_error());

mysql\_select\_db($my\_database) or die('Could not select database');

$number = $\_POST[number];

$description = $\_POST[description];

$price = $\_POST[price];

$client = $\_POST[client];

$date\_pr = $cur\_day;

$date\_dog = $\_POST[date\_dog];

$date\_fact = $\_POST[date\_fact];

$master = $\_POST[master];

//adding to database

if ($number <> null and $description <> null and $price <> null and $client <> null and $date\_pr <> null and $date\_dog <> null and $master <> null)

{

$query = "INSERT INTO `Order` (number, description, price, client, date\_pr, date\_dog, date\_fact, master)

VALUES ($number, '$description', $price, '$client', '$date\_pr', '$date\_dog', '$date\_fact', '$master')";

$sql = mysql\_query ($query) or die('Could not write database: ' . mysql\_error());

}

//adding done order

if ($\_POST[done\_number] <> null and $\_POST[done\_date\_fact] <> null)

{

$query = "UPDATE `Order` SET date\_fact = '$\_POST[done\_date\_fact]' WHERE number = '$\_POST[done\_number]'";

$result = mysql\_query($query) or die('Query failed: ' . mysql\_error());

}

$query = "SELECT \* FROM `Order` WHERE date\_pr = '".$cur\_day."'";

$result = mysql\_query($query) or die('Query failed: ' . mysql\_error());

echo "<b>Список принятых сегодня заказов:

<form action='' method='POST'>

<table border='1' width='100%' cellpadding='0'>

<tr>

<td width='125'> <p align='center'>№</td>

<td width='125'> <p align='center'>Описание</td>

<td width='125'> <p align='center'>Цена</td>

<td width='125'> <p align='center'>Клиент</td>

<td width='125'> <p align='center'>Дата приема</td>

<td width='125'> <p align='center'>Дата по договору</td>

<td width='125'> <p align='center'>Дата фактическая</td>

<td width='125'> <p align='center'>Мастер</td>

</tr>

</table>

</form> ";

echo "<table border='1' width='100%' cellpadding='0'>";

while ($row = mysql\_fetch\_array ($result, MYSQL\_ASSOC))

{ echo "<tr>";

foreach ($row as $value)

{ echo "<td width='125'>$value</td>";

}

echo "</tr>";

}

echo "</table>";

$query = "SELECT price FROM `Order` WHERE date\_pr = '".$cur\_day."'";

$result = mysql\_query($query) or die('Query failed: ' . mysql\_error());

while ($row = mysql\_fetch\_array ($result, MYSQL\_ASSOC))

{ foreach ($row as $value)

{ $profit = $profit + $value;

}

}

echo "Выручка: ".$profit." руб.";

?>

## Исходный текст модуля Orders.php

<?php

$mysql\_host = "localhost";

$my\_database = "data";

$mysql\_user = "root";

echo "<h5 align='center'>Система для автоматизации обработки запросов заказов на швейном предприятии v.1.0</h5>";

echo "<h3 align='center'>Незавершенные заказы</h3>";

echo "<a href = 'index.php'> Оформление заказов</a> ";

echo "<a href = 'orders.php'> Незавершенные заказы</a> ";

echo "<a href = 'search.php'> Поиск по заказам</a><br><br>";

$cur\_day = date("Y.m.d");

echo "<b>Текущее число: ".$cur\_day."<br><br>";

// Connecting, selecting database

$link = mysql\_connect($mysql\_host, $mysql\_user)

or die('Could not connect: ' . mysql\_error());

//echo 'Connected successfully<br>';

mysql\_select\_db($my\_database) or die('Could not select database');

$query = "SELECT number, description, date\_dog, master FROM `Order` WHERE date\_fact = '0000-00-00' ORDER BY `order`.`date\_dog` ASC";

$result = mysql\_query($query) or die('Query failed: ' . mysql\_error());

echo "<form action='' method='POST'>

<table border='1' width='50%' cellpadding='0'>

<tr>

<td width='125'> <p align='center'>№</td>

<td width='125'> <p align='center'>Описание</td>

<td width='125'> <p align='center'>Дата по договору</td>

<td width='125'> <p align='center'>Мастер</td>

</tr>

</table>

</form> ";

echo "<table border='1' width='50%' cellpadding='0'>";

while ($row = mysql\_fetch\_array ($result, MYSQL\_ASSOC))

{ echo "<tr>";

foreach ($row as $value)

{ echo "<td width='125'>$value</td>";

}

echo "</tr>";

}

echo "</table>";

?>

## Исходный текст модуля Search.php

<?php

$mysql\_host = "localhost";

$my\_database = "data";

$mysql\_user = "root";

echo "<h5 align='center'>Система для автоматизации обработки запросов заказов на швейном предприятии v.1.0</h5>";

echo "<h3 align='center'>Поиск по заказам</h3>";

echo "<a href = 'index.php'> Оформление заказов</a> ";

echo "<a href = 'orders.php'> Незавершенные заказы</a> ";

echo "<a href = 'search.php'> Поиск по заказам</a><br><br>";

$cur\_day = date("Y.m.d");

echo "<b>Текущее число: ".$cur\_day."<br><br>";

// Connecting, selecting database

$link = mysql\_connect($mysql\_host, $mysql\_user)

or die('Could not connect: ' . mysql\_error());

//echo 'Connected successfully<br>';

mysql\_select\_db($my\_database) or die('Could not select database');

echo "<b>Введите критерии поиска:<br>";

echo "<form action='' method='POST'>

<table border='1' width='100%' cellpadding='0'>

<tr>

<td> <p align='center'>№</td>

<td> <p align='center'>Описание</td>

<td> <p align='center'>Цена</td>

<td> <p align='center'>Клиент</td>

<td> <p align='center'>Дата приема</td>

<td> <p align='center'>Дата по договору</td>

<td> <p align='center'>Дата фактическая</td>

<td> <p align='center'>Мастер</td>

</tr>

<tr>

<td> <p align='center'><input type='text' name='number'></td>

<td> <p align='center'><input type='text' name='description'></td>

<td> <p align='center'><input type='text' name='price'></td>

<td> <p align='center'><input type='text' name='client'></td>

<td> <p align='center'><input type='text' name='date\_pr'></td>

<td> <p align='center'><input type='text' name='date\_dog'></td>

<td> <p align='center'><input type='text' name='date\_fact'></td>

<td> <p align='center'><input type='text' name='master'></td>

</tr>

</table>

<input type='submit' value='Поиск'>

</form";

if ($\_POST[number] == null) {$number = "";}

else $number = "number = $\_POST[number] AND ";

if ($\_POST[description] == null) {$description = "";}

else $description = "description LIKE '".$\_POST[description]."' AND ";

echo $description;

if ($\_POST[price] == null) {$price = "";}

else $price = "price = $\_POST[price] AND ";

if ($\_POST[client] == null) {$client = "";}

else $client = "client LIKE'".$\_POST[client]."' AND ";

if ($\_POST[date\_pr] == null) {$date\_pr = "";}

else $date\_pr = "date\_pr ='".$\_POST[date\_pr]."' AND ";

if ($\_POST[date\_dog] == null) {$date\_dog = "";}

else $date\_dog = "date\_dog ='".$\_POST[date\_dog]."' AND ";

if ($\_POST[date\_fact] == null) {$date\_fact = "";}

else $date\_fact = "date\_fact ='".$\_POST[date\_fact]."' AND ";

if ($\_POST[master] == null) {$master = "";}

else $master = "master LIKE'".$\_POST[master]."' AND ";

$end\_line = "1";

$query = "SELECT \* FROM `Order` WHERE $number $description $price $client $date\_pr $date\_dog $date\_fact $master $end\_line";

$result = mysql\_query($query) or die('Query failed: ' . mysql\_error());

echo "<br><br><br><b>Найденные заказы:";

echo "<form action='' method='POST'>

<table border='1' width='100%' cellpadding='0'>

<tr>

<td width='125'> <p align='center'>№</td>

<td width='125'> <p align='center'>Описание</td>

<td width='125'> <p align='center'>Цена</td>

<td width='125'> <p align='center'>Клиент</td>

<td width='125'> <p align='center'>Дата приема</td>

<td width='125'> <p align='center'>Дата по договору</td>

<td width='125'> <p align='center'>Дата фактическая</td>

<td width='125'> <p align='center'>Мастер</td>

</tr>

</table>

</form> ";

echo "<table border='1' width='100%' cellpadding='0'>";

while ($row = mysql\_fetch\_array ($result, MYSQL\_ASSOC))

{ echo "<tr>";

foreach ($row as $value)

{ echo "<td width='125'>$value</td>";

}

echo "</tr>";

}

echo "</table>";

//echo "<br>"."<br>";

//echo "<br>".$query;

?>

## Исходный текст модуля Order.sql

-- phpMyAdmin SQL Dump

-- version 3.2.3

-- http://www.phpmyadmin.net

--

-- Host: localhost

-- Generation Time: Feb 13, 2012 at 01:48 PM

-- Server version: 5.1.40

-- PHP Version: 5.3.1

SET SQL\_MODE="NO\_AUTO\_VALUE\_ON\_ZERO";

--

-- Database: `data`

--

-- --------------------------------------------------------

--

-- Table structure for table `order`

--

CREATE TABLE IF NOT EXISTS `order` (

`number` int(11) NOT NULL,

`description` text NOT NULL,

`price` int(11) NOT NULL,

`client` text NOT NULL,

`date\_pr` date NOT NULL,

`date\_dog` date NOT NULL,

`date\_fact` date NOT NULL,

`master` text NOT NULL,

KEY `number` (`number`),

KEY `number\_2` (`number`),

KEY `number\_3` (`number`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251;

--

-- Dumping data for table `order`

--

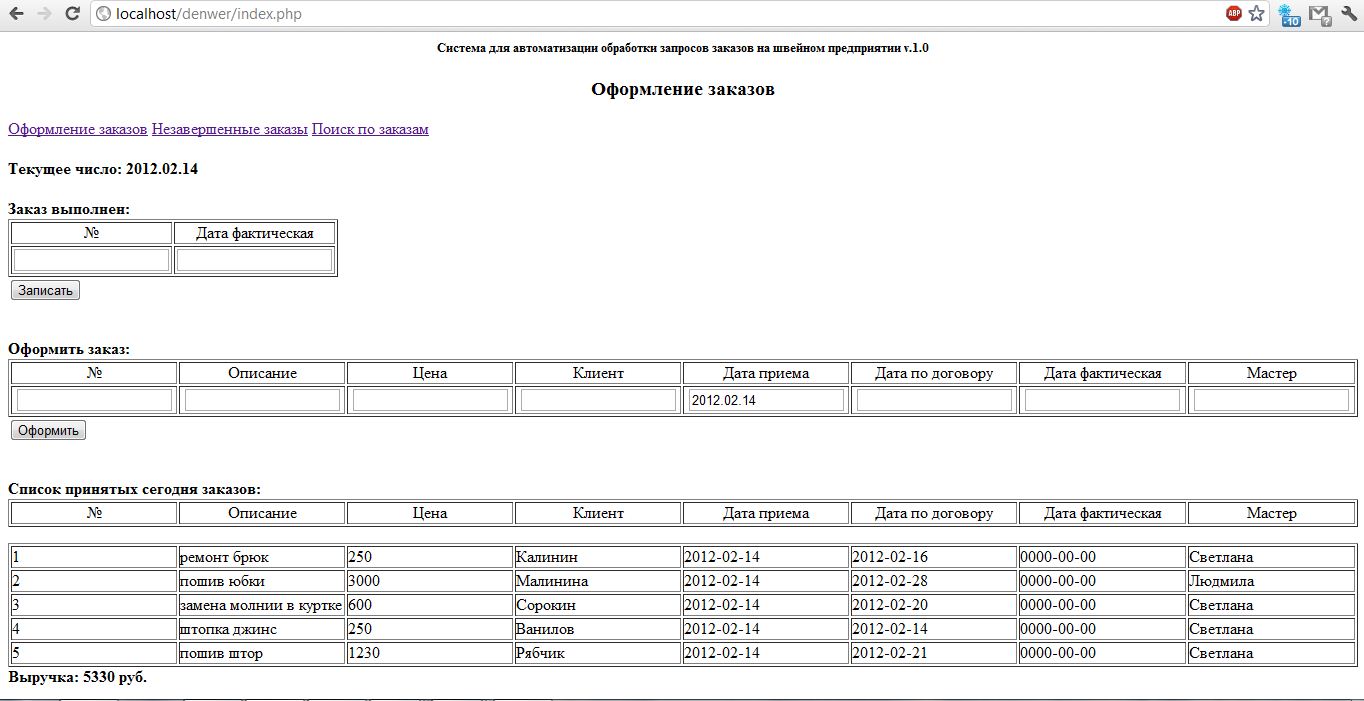
# ПРИЛОЖЕНИЕ 2

Модуль оформление заказов (index.php). Начальное состояние.

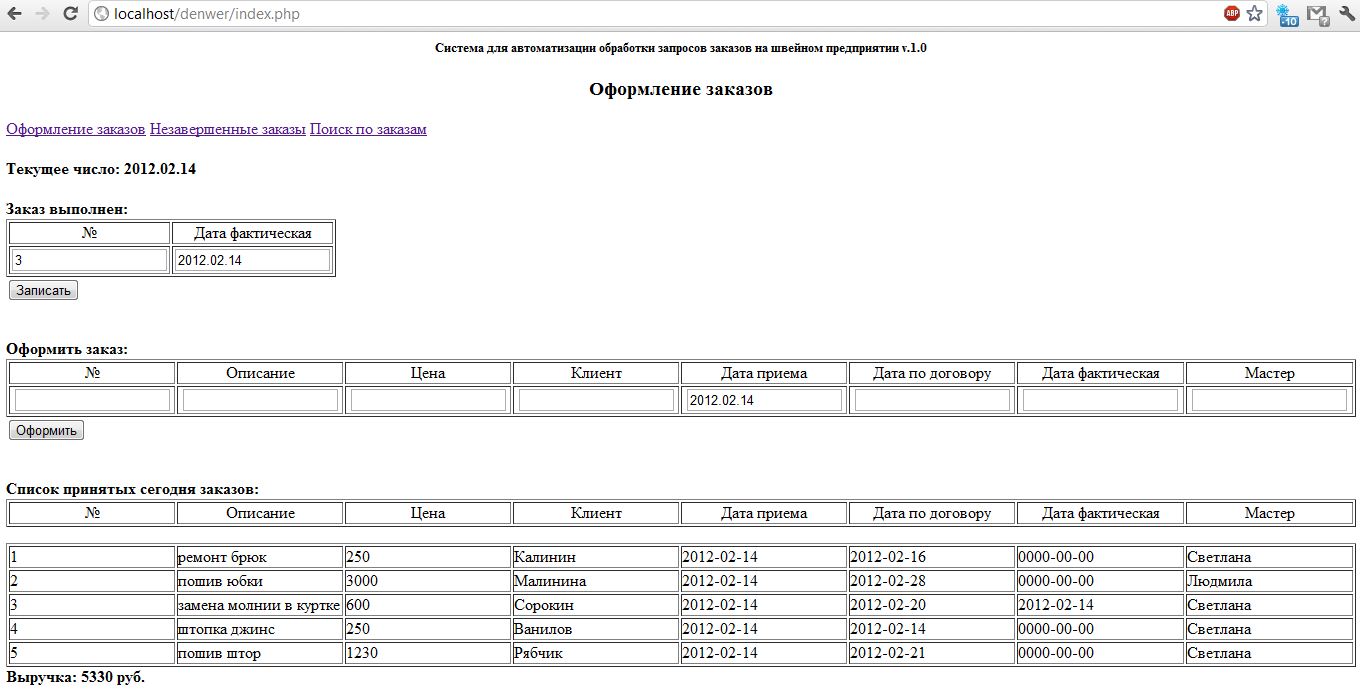
 Модуль оформление заказов (index.php). Процесс оформления заказа.



Модуль оформление заказов (index.php). Список оформленных сегодня заказов.



Модуль оформление заказов (index.php). Запись информации о сделанных заказах.



Модуль незавершенные заказы (orders.php)



Модуль поиск по заказам (search.php)





