

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Построение модели данных	5
1.1 Работа по методическим указаниям	5
1.1.1 Индивидуальное задание	7
2 Создание логической и физической модели данных	10
2.1 Работа по методическим указаниям	10
2.1.1 Индивидуальное задание	12
3 Конструкторский раздел	15
3.1 Архитектура всячины	15
3.2 Подсистема всякой ерунды	15
3.2.1 Блок-схема всякой ерунды	15
4 Технологический раздел	17
5 Экспериментальный раздел	19
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21
А Картинки	22
Б Еще картинки	23

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Распределённый — Слово, которое нельзя употреблять. Но надо протестировать длинные строки в глоссарии.

ПЕРЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

SQL — Structured query language.

БД — База данных.

ПО — Программное обеспечение.

СУБД — Система управления базами данных.

ВВЕДЕНИЕ

Целью работы является построение логической и физической модели данных выбранной предметной области, освоение навыков работы с программным обеспечением ERwin Data Modeler, ErAssistant, MySQL Workbench, MySQL Server.

На заключительном этапе работы необходимо представить веб–приложение или полнофункциональное приложение, организующее взаимодействие пользователя с созданной базой данных.

1 Построение модели данных

В данном разделе описывается и создается модель данных предложенной предметной области в ПО ErAssistant.

1.1 Работа по методическим указаниям

Произведем описание предметной области, рассматриваемой в методических указаниях.

Описание предметной области

Компания занимается производством и продажей небольших статуэток, раскрашиваемых вручную. Компания имеет несколько производственных направлений. Миниатюры изготавливаются из гипса, олова или алюминия.

Компания распространяет свои товары по трем каналам. Компания содержит пять собственных розничных магазинов. Помимо этого, компания владеет сайтом, на котором ведется online-торговля, и осуществляет оптовые поставки сторонним дистрибьюторам. Для анализа статистики, системой автоматизации производства, нужен интерактивный аналитический инструмент. Поэтому необходимо спроектировать и построить модель данных, которая станет хранилищем информации по производству.

В ходе производства изделий система автоматизации производства управляет всеми станками компании. Каждый станок реализует полный цикл производства изделий, включая:

- заполнение формы сырьем (гипсом, оловом или алюминием);
- затвердевание материала;
- удаление изделия из формы после затвердевания;
- при необходимости автоматизированная раскраска изделий (оловянные фигурки не раскрашиваются);
- сушку после покраски (при необходимости).

Покраска и сушка могут производиться за несколько этапов в зависимости от сложности изделия. По мере готовности изделия проходят проверку, выполняемую оператором станка.

Оператор станка регистрируется в системе. В ходе этого процесса оператор сообщает системе автоматизации производства тип производимых изделий и объем загруженного в машину сырья. Оператор также делает в системе запись при отбраковке изделий.

В ходе интервью необходимые для эффективного анализа статистики:

- число принятых изделий по объему сырья, видам изделий, машинам и
- время формовки и затвердевания по видам изделий, машинам и дням;
- время покраски и сушки по типам краски, видам изделий, машинам и
- сворачивание по подтипам изделий, которые сворачиваются по типам;
- сворачивание по типам машин, которые сворачиваются по материалам (гипс, олово или алюминий);
- сворачивание машин по фабрикам, которые сворачиваются по странам;
- сворачивание дней по месяцам, месяцев — по кварталам;
- возможность фильтрации информации по производителю и дате покупки машины.

Анализ файла-экспорта из системы автоматизации производства показал, что для каждого вида производимых изделий есть отдельная строка, в которой присутствует следующая информация:

- тип изделия;
- объем сырья;
- номер машины;
- личный номер оператора;
- время и дата начала производства (когда серия начата);
- время и дата окончания производства (когда серия закончена);
- флаг отбраковки.

Построение модели

По приведенному описанию предметной области построим ее модель в среде ErAssistant . Укажем линии связей, назначим им имена, укажем типы и кратность связей. В результате работы, модель примет вид, приведенный на Рисунке 1.1.

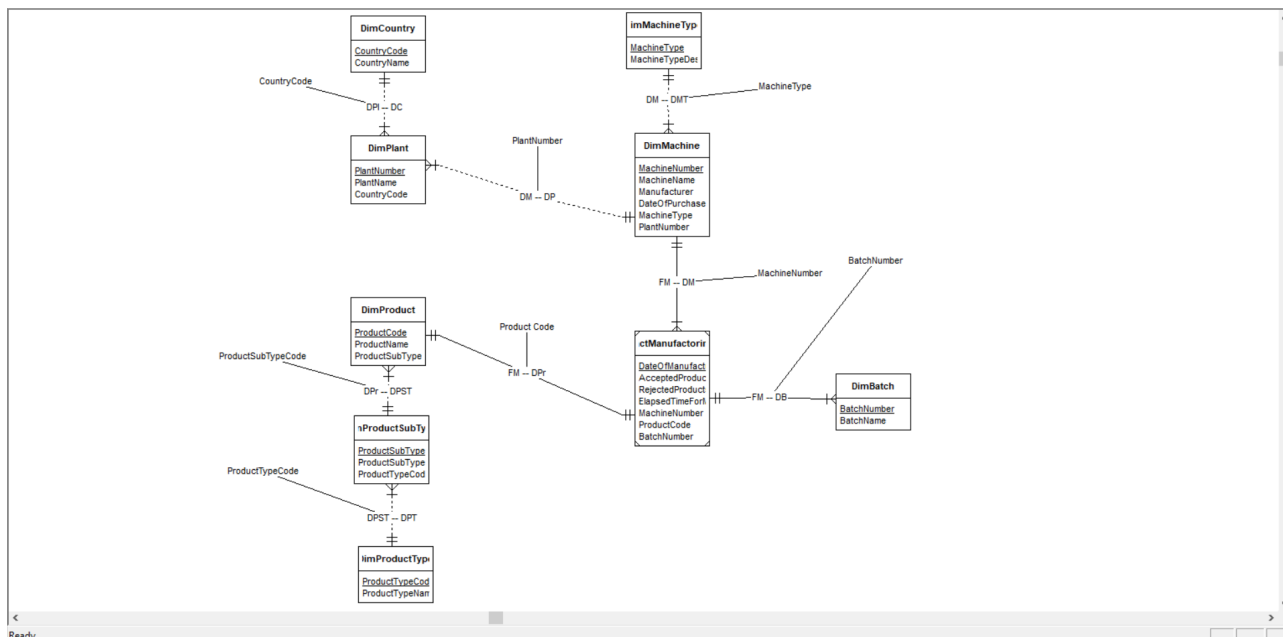


Рисунок 1.1 — Модель данных Производства со связями

1.1.1 Индивидуальное задание

Описание предметной области

В качестве индивидуального задания была выбрана реализация модели информационной системы по хранению и анализу данных предприятия, занимающегося сборкой и поставкой спортивных велосипедов для конечных потребителей по индивидуальному заказу.

Предприятие располагает широким выбором компонентов и комплектующих для сборки велосипедов следующих типов:

- дорожный;
- горный;
- кросс-кантри;
- эндуро;
- Прогулочный.

Предприятие может предложить сконфигурировать велосипед, отдельно выбрав каждый из предложенных компонентов:

- рама;
- вилка;
- руль;

- трансмиссия;
- колеса;
- тормозная система.

Контроль над выполнением работ по сборке велосипеда проводится в виде учета всех операций по сборке, настройке и тестированию, проводимых на территории предприятия ее сотрудниками. При этом каждая запись содержит следующую информацию о проведенных работах:

- внутренний номер изделия;
- время;
- этап работ;
- название цеха;
- имя мастера;
- статус;
- примечание.

На предприятии ведется учет всех компонентов велосипедов. В базе данных предприятия хранится информация о каждом компоненте, приобретенном у партнеров или изготовленном самостоятельно.

В независимости от типа компонента он обладает общей информацией о наименовании производителя, месте и времени изготовления, типе и рекомендованной розничной цене. Также каждый компонент имеет особые сведения, присущие данному типу детали.

Построение модели

После приведения общих сведений о роде деятельности предприятия, факторизируем модель данных информационной системы предприятия в среде ErAssistant (см Рисунок 1.2). Приложение ErAssistant позволяет пользователю создавать, редактировать диаграммы сущностей и связей.

Укажем названия связей, их идентификаторы и кратность, исходя из вида отношений, выстроенных между сущностями.

На данном этапе выполнения работы мы реализовали модель данных будущего хранилища данных, планируемого к применению в компании, занимающейся сборкой и поставкой велосипедов по индивидуальному заказу. Был

2 Создание логической и физической модели данных

В данном разделе будет рассмотрено создание логической и физической модели данных предложенных предметных областей в ПО ERwin Data Modeler.

На этапе инфологического проектирования базы данных должна быть построена модель предметной области, не привязанная к конкретной СУБД, понятная не только разработчикам информационной системы, но и экономистам, менеджерам и другим специалистам. В то же время модель предметной области должна максимально точно отражать семантику предметной области и позволять легко перейти к модели данных конкретной СУБД.

Логический уровень — это уровень, соответствующий инфологическому этапу проектирования и не привязанный к конкретной СУБД. Модели логического уровня оперируют с понятиями сущностей, атрибутов и связей, которые на этом уровне именуются на естественном языке (в нашем случае — на русском) так, как они называются в реальном мире.

Физический уровень — это отображение логической модели на модель данных конкретной СУБД. Одной логической модели может соответствовать несколько физических моделей. Причем, Erwin (как и другие CASE-системы проектирования баз данных) позволяет автоматизировать отображение логической модели на физическую.

2.1 Работа по методическим указаниям

Порядок построения модели данных в среде ERwin Data Modeler рассмотрим на примере автоматизированной информационной системы «Реализация средств вычислительной техники», предназначенной для учета продаж настольных компьютеров по заказам клиентов.

Создание модели данных начинается с разработки логической модели, которая должна представлять состав сущностей предметной области с перечнем атрибутов и отношений между ними.

Результат разработки логической модели данных системы «Реализация средств вычислительной техники», предназначенной для учета продаж настольных компьютеров по заказам клиентов приведен на Рисунке 2.1.

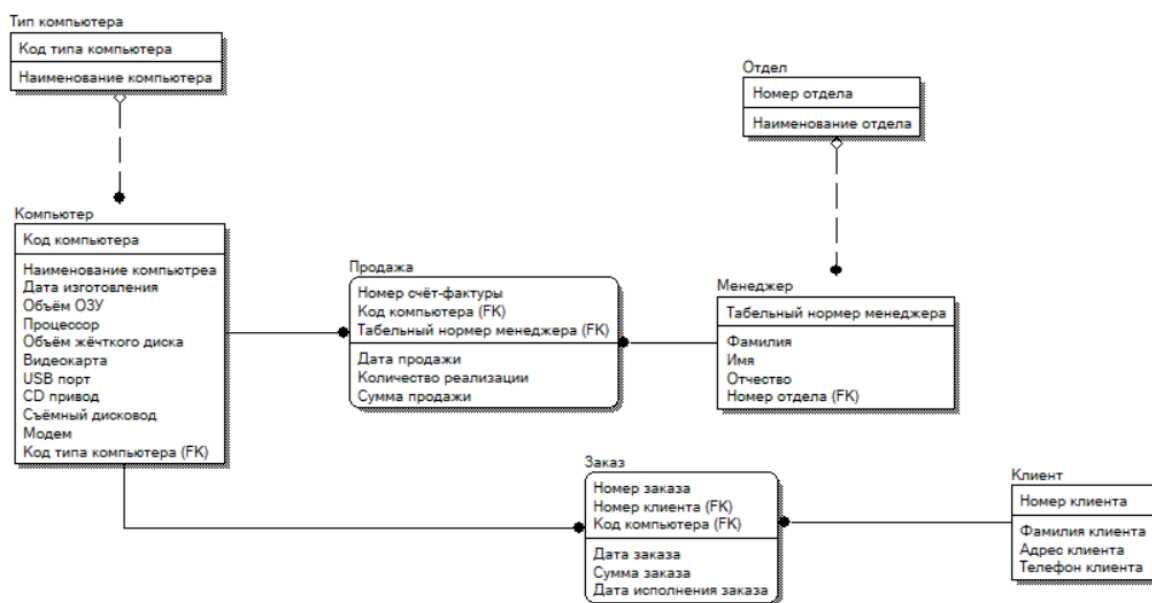


Рисунок 2.1 — Логическая модель данных системы «Реализация средств вычислительной техники»

Для построения физической модели данных системы, следует определиться с СУБД, в которой будет реализована модель. При построении физической модели данных следует учитывать формальную теория представления и обработки данных в конкретной системе управления базами данных (СУБД).

В данной практической работе в качестве СУБД выбрана MySQL.

Приступим к построению физической модели данных системы «Реализация средств вычислительной техники». Результат работы можно видеть на Рисунке 2.2.

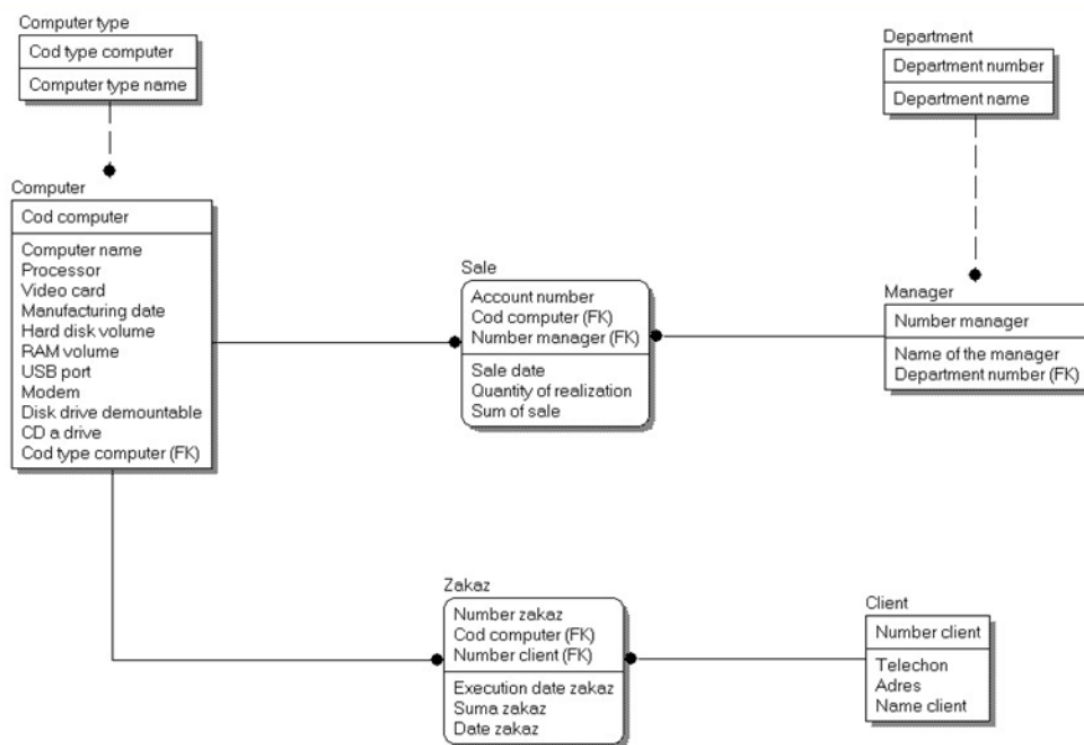


Рисунок 2.2 — Физическая модель данных системы «Реализация средств вычислительной техники»

2.1.1 Индивидуальное задание

Приступим к построению логической модели данных системы «Велосипедное предприятие». В соответствии с моделью, реализованной в ходе первой практической работы, добавим в рабочую область следующие сущности:

- Component;
- FrameInfo;
- Frame;
- Frameset;
- FrameSize;
- Fork;
- ComponentType;
- Wheelset;
- Groupset;
- Brake;
- FctCycleBuild;
- CycleType;

- Bar;
- Setup.

Добавим связи между сущностями в соответствии с ранее построенной моделью. Логическая модель системы «Велосипедное предприятие» приведена на Рисунке 2.3.

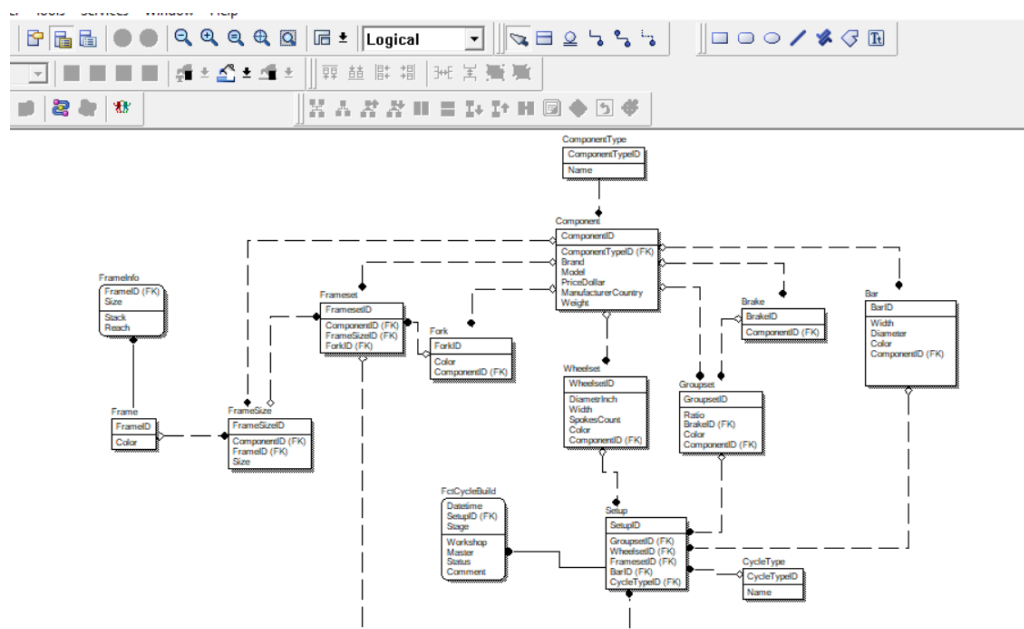


Рисунок 2.3 — Логическая модель данных системы «Велосипедное предприятие»

После уточнения типов данных, выбранных в соответствии с предметной областью и спецификой СУБД MySQL. Физическая модель системы «Велосипедное предприятие» приведена на Рисунке 2.4.

После реализации физической и логической модели можно приступать к реализации модели данной системы в СУБД MySQL.

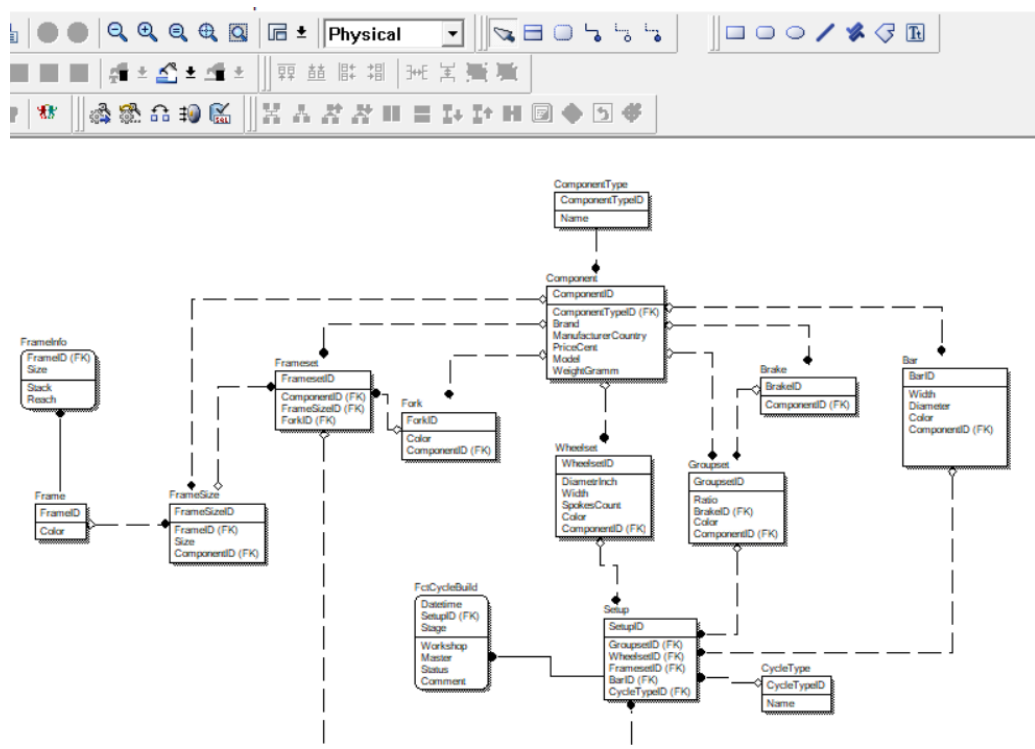


Рисунок 2.4 — Физическая модель данных системы «Велосипедное предприятие»

3 Конструкторский раздел

В данном разделе проектируется новая всячина.

3.1 Архитектура всячины

Проверка параграфа. Вроде работает.

Вторая проверка параграфа. Опять работает.

Вот.

- Это список с «палочками».
- Хотя он и не по ГОСТ, кажется.

а) Поэтому для списка, начинающегося с заглавной буквы, лучше список с цифрами.

Формула 3.1 совершенно бессмысленна.

$$a = cb \tag{3.1}$$

Окружение cases опять работает (см. 3.2), спасибо И. Короткову за исправления..

$$a = \begin{cases} 3x + 5y + z, & \text{если хорошо} \\ 7x - 2y + 4z, & \text{если плохо} \\ -6x + 3y + 2z, & \text{если совсем плохо} \end{cases} \tag{3.2}$$

3.2 Подсистема всякой ерунды

Культурная вставка dot-файлов через утилиту dot2tex (рис. 3.1).

3.2.1 Блок-схема всякой ерунды

Кстати о заголовках

У нас есть и **subsubsection**. Только лучше её не нумеровать.

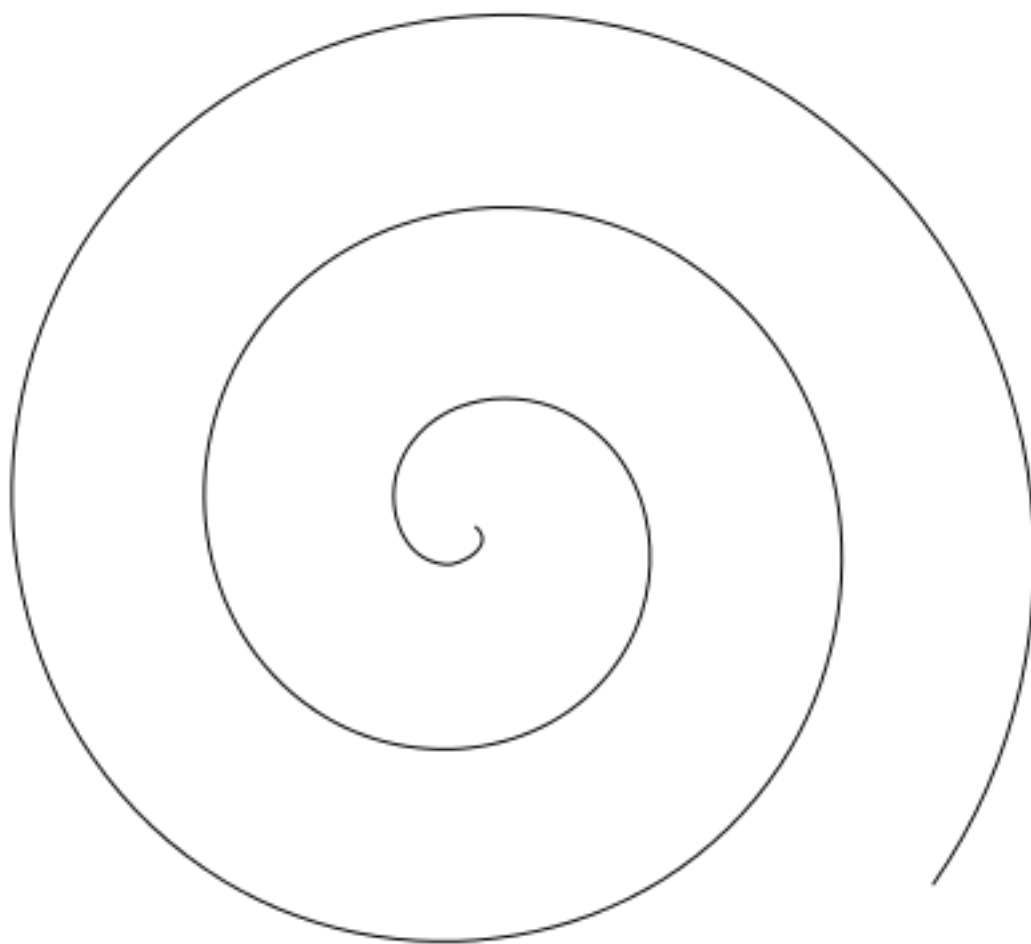


Рисунок 3.1 — Рисунок

4 Технологический раздел

В данном разделе описано изготовление и требование всячины. Кстати, в Latex нужно эскейпить подчёркивание (писать «some_function» для **some_function**).

Для вставки кода есть пакет **listings**. К сожалению, пакет **listings** всё ещё работает криво при появлении в листинге русских букв и кодировке исходников utf-8. В данном примере он (увы) на лету конвертируется в koi-8 в ходе сборки pdf.

Есть альтернатива **listingsutf8**, однако она работает лишь с **\lstinputlisting**, но не с окружением **\lstlisting**

Вот так можно вставлять псевдокод (питоноподобный язык определен в **listings.inc.tex**):

Листинг 4.1 — Алгоритм оценки дипломных работ

```
1 def EvaluateDiplomas():
2     for each student in Masters:
3         student.Mark ← 5
4     for each student in Engineers:
5         if Good(student):
6             student.Mark ← 5
7         else:
8             student.Mark ← 4
```

Еще в шаблоне определен псевдоязык для BNF:

Листинг 4.2 — Грамматика

```
1 ifstmt → "if" "(" expression ")" stmt |
2         "if" "(" expression ")" stmt1 "else" stmt2
3 number → digit digit *
```

В листинге 4.3 работают русские буквы. Сильная магия. Однако, работает только во включаемых файлах, прямо в TeX нельзя.

Листинг 4.3 — Пример (test.c)

```
1 #include <stdio.h>
2 int main()
3 {
4     return 0;
5 }
```

Можно также использовать окружение **verbatim**, если **listings** чем-то не устраивает. Только следует помнить, что табы в нём «съедаются». Существует так же команда **\verbatiminput** для вставки файла.

```
a_b = a + b; // русский комментарий
if (a_b > 0)
    a_b = 0;
```

5 Экспериментальный раздел

В данном разделе проводятся вычислительные эксперименты. А на рис. 5.1 показана схема мыслительного процесса автора...

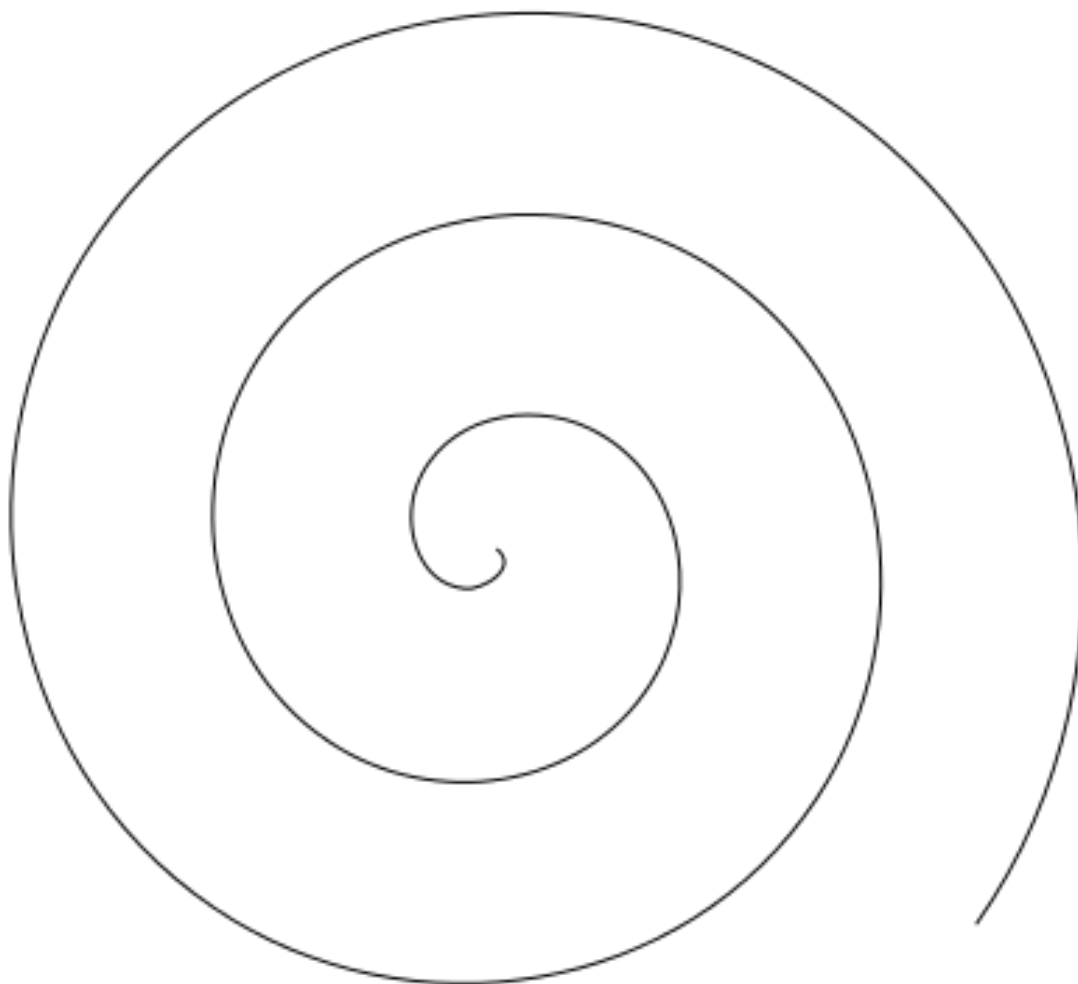


Рисунок 5.1 — Как страшно жить

ЗАКЛЮЧЕНИЕ

В результате проделанной работы стало ясно, что ничего не ясно...

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Дюбуа, Поль*. MySQL: Полн. и исчерпывающее руководство по применению и администрированию баз данных MySQL 4, а также программированию приложений / Поль Дюбуа. — Издательский дом Вильямс, 2004.
2. *Юртанова, Екатерина Михайловна*. Разработка Web-приложений с использованием языка PHP / Екатерина Михайловна Юртанова // *Учебный эксперимент в образовании*. — по. 1.
3. *Балдин, Евгений Михайлович*. Компьютерная типография LaTeX / Евгений Михайлович Балдин. — Evgeny Baldin, 2008.

Приложение А Картинки

Рисунок А.1 — Картинка в приложении. Страшная и ужасная.

Приложение Б Еще картинки

Рисунок Б.1 — Еще одна картинка, ничем не лучше предыдущей. Но надо же как-то заполнить место.