

Содержание

| | |
|--|----|
| Введение | 2 |
| 1 Аналитический раздел | 3 |
| 1.1 Описание предметной области | 3 |
| 1.2 Определение требований к структуре базы данных | 5 |
| 1.2.1 Определение целей создания системы | 5 |
| 1.2.2 Определение объема и типов данных | 5 |
| 1.2.3 Определение способа использования данных | 6 |
| 1.2.4 Определение бизнес-правил | 7 |
| 1.3 Разработка логической модели данных | 7 |
| 1.3.1 Отпределение сущностей, связей между ними и атрибутов сущностей | 7 |
| 1.3.2 Определение ограничений налагаемых на данные | 7 |
| 1.3.3 Диаграмма БД | 7 |
| 2 Конструкторский раздел | 9 |
| 2.1 Архитектура всячины | 9 |
| 2.2 Подсистема всякой ерунды | 9 |
| 2.2.1 Блок-схема всякой ерунды | 9 |
| 3 Технологический раздел | 11 |
| 4 Экспериментальный раздел | 13 |
| Заключение | 14 |
| Список использованных источников | 15 |

Введение

В настоящее время существует большое разнообразие социальных сетей различных направлений. Основными потребностями пользователей данных продуктов являются:

- Общение по средством сообщений, комментариев...
- Получение информации из определенных источников (публикаций других пользователей)

Успешными социальными сетями, как правило, пользуются несколько миллионов человек ежедневно и десятки миллионов пользователей в месяц. Количество зарегистрированных пользователей может достигать сотен миллионов. Активные пользователи периодически создают данные (сообщения, посты, комментарии...). Разумеется, что такой объем данных физически не может уместиться на одном сервере. К тому же пользователи, как правило, активно осуществляют доступ к контенту, созданному другими пользователями. Как правило этот контент представляется в виде “лент” - последовательностей событий, агрегированных и отсортированных различным образом и изменяющихся во времени.

Целью данного курсового проекта является создание новостной социальной сети, количество серверов для обеспечения работы которой можно было бы линейно увеличивать в зависимости от числа пользователей. Так как социальная сеть состоит из достаточно большого числа компонентов, реализация полностью масштабируемой системы является сложной задачей, выходящей за рамки курсового проекта. Требования масштабируемости предъявляются только для самих страниц лент и текста публикаций в этих лентах.

Для достижения поставленной цели необходимо решить следующие задачи

- провести аналитическое исследование предметной области;
- сконфигурировать все используемые хранилища данных;
- создать веб-приложение, осуществляющее доступ к данным.

1 Аналитический раздел

Целью данного раздела является аналитическое исследование предметной области. Для этого необходимо:

- подробно описать предметную область;
- определить требования к структуре базы данных;
- разработать логическую модель данных.

1.1 Описание предметной области

Новостная социальная сеть предоставляет следующую функциональность:

- регистрация пользователей;
- публикация сообщений, содержащих текст и изображение(не обязательно);
- повторная публикация сообщений, ранее опубликованных другими пользователями;
- комментирование сообщений;
- выражение своего отношения к публикации посредством кнопок “нравится” и “не нравится”;
- отображение списков публикаций

Каждый пользователь при регистрации может указать свои имя, фамилию, “ник”, некоторое текстовое описание в произвольной форме, пол, дату рождения. Так же пользователь может установить изображение в качестве “аватара”. Аватары пользователя отображаются на его странице и рядом с его публикациями. Вся эта информация может быть изменена пользователем, если это допускается ограничениями. Ник является уникальным идентификатором пользователя и используется для формирования адреса его страницы. При регистрации аккаунт пользователя привязывается к некоторому адресу электронной почты. Так же пользователь указывает пароль при регистрации. Обязательными для пользователя являются только “ник”, электронный адрес и пароль. В случае если аватар пользователя не задан, то отображается некоторое изображение по умолчанию. Пользователи получают доступ в систему при помощи адреса электронной почты и пароля. Пароль пользователя не может быть сохранен в БД в прямом виде, и должен отвечать некоторым требованиям сложности. Пароль может быть сброшен пользователем, если он забыл его, в этом случае пользователю на его электронный адрес будет отправлено письмо, содержащее ссылку для сброса пароля, пароль сбрасывается только при переходе по этой ссылке и успешном вводе нового пароля. Если пользователь не завершил эти действия, старый пароль остается в силе. Ссылка для сброса пароля действительна только 1 день, и может быть использована только один раз. Пользователь может получить новую ссылку для сброса пароля, только если не существует еще дейст-

тельной другой ссылки сброса пароля. Если пользователь успешно авторизуется при помощи текущего логина и пароля, то действительная ссылка для сброса пароля объявляется недействительной. Так же пользователь может сменить пароль на странице анкеты, при помощи текущего пароля. Ссылки для сброса пароля содержат некоторый константный префикс и идентификатор ссылки сброса(далее “токен”). Токен представляет собой некоторую строку состоящую из букв латинского алфавита различного регистра и цифр длиной 30 символов. Электронный адрес пользователя доступен для чтения только самому пользователю и не может быть изменен.

Пользователи могут создавать публикации, содержащие текст и изображение, ограниченные по размеру. В будущем возможно увеличение числа изображений содержащихся в одной публикации.

Пользователи могут подписываться на публикации других пользователей, выражать свое отношение к публикации по средством кнопок “нравится” и “не нравится”.

Для каждой публикации выводиться число пользователей выразивших свое отношение в каждой из категорий “нравится” и “не нравится”. Пользователь одновременно может выразить свое отношение только в одной из категории только один раз.

Публикации предоставляются пользователям в виде “лент”. Существует два типа лент: лента публикаций конкретного пользователя и агрегированная лента.

Лента публикаций конкретного пользователя представляет собой отсортированную по времени последовательность публикаций этого пользователя и повторных публикаций, сделанных им.

Агрегированная лента составляется как отсортированная по времени последовательность публикаций и повторных публикаций пользователей, на которых подписан текущий пользователь.

Публикации выводятся с прикрепленными к ним счетчиками “нравится” и “не нравится”, комментариями, датой и временем создания. Рядом с публикациями и повторными публикациями всегда выводится аватар и ник пользователя, который опубликовал или повторно опубликовал данное сообщение соответственно. Рядом с повторными публикациями дополнительно выводится аватар и ник пользователя, который первым опубликовал данное сообщение.

Пользователи могут повторно опубликовывать любое сообщение посредством одного нажатия кнопки мыши.

Комментарии выводятся в отсортированном по времени создания виде, последние 2 комментария выводятся всегда, остальные могут быть спрятаны, но должны быть доступны. Для каждого комментария указываются дата создания, ник и аватар его автора.

Большую часть времени пользователи подобных социальных сетей проводят посещая страницы других пользователей и читая агрегированную ленту.

Страница пользователя будет содержать:

- аватар пользователя;
- анкетные данные пользователя;
- ленту его публикаций.

Для агрегированной ленты пользователя так же должна быть создана отдельная страница.

1.2 Определение требований к структуре базы данных

1.2.1 Определение целей создания системы

Основной целью новостной социальной сети является предоставление пользователям возможности обмениваться новостями(опубликованными сообщениями). Новостная лента является достаточно удобным способом получения информации от разных источников, а новостная социальная сеть позволяет также удобным образом настраивать ленту с помощью системы подписок. Пользователи смогут подписываться на публикации своих друзей и других интересных им пользователей.

1.2.2 Определение объема и типов данных

Исходя из анализа предметной области можно выделить следующие категории данных:

- пользователи
- публикации
- комментарии
- изображения
- отметки “нравится” и “не нравится”
- списки подписок пользователя на других пользователей

Для категории пользователей необходимо хранить следующие сведения:

- электронный адрес, к которому привязан аккаунт данного пользователя;
- пароль(в искаженном виде);
- токен сброса пароля;
- ник;
- аватар;
- имя;
- фамилия;
- дата рождения;

- пол;
- описание.

Для категории публикаций:

- текст;
- изображение;
- автор;
- время создания.

Для

Ожидаемое количество пользователей системы не превышает 100 000 000 человек. Имя пользователя, его ник и фамилия не могут быть более двадцати символов, электронный адрес пользователя не может быть более 50-и символов.

Ожидаемое среднее число подписок на одного пользователя не превышает трехсот.

Предполагается, что все пользователи или наибольшая их часть будут иметь аватары. Аватар - изображение в формате jpeg, не превышающее по размеру 10 MiB.

Предполагается, что в среднем пользователи будут создавать одну публикацию в неделю. Текст публикации не может быть больше 1000 символов. Изображение прикрепленное к публикации так же как и аватар должно быть в формате jpeg и не превышать по размеру 10 MiB. Для каждой публикации хранится время ее создания в формате unix-time. Ожидается, что в среднем на одну публикацию будет приходиться одна повторная публикация.

Комментарии к постам представляют из себя текст не превышающий 500 символов. Для комментариев так же хранится время их создания. Предполагается, что в среднем к каждой публикации будет прикреплен 1 комментарий.

Ожидаемое число отметок “нравится” и “не нравится” - 10 на каждую публикацию.

Срок жизни проекта - 5 лет.

1.2.3 Определение способа использования данных

Каждый пользователь может одновременно выступать в двух ролях:

- постовщика контента;
- читателя новостной ленты.

1.2.4 Определение бизнес-правил

1.3 Разработка логической модели данных

1.3.1 Определение сущностей, связей между ними и атрибутов сущностей

1.3.2 Определение ограничений налагаемых на данные

1.3.3 Диаграмма БД

На рисунке [?] представлена ER-диаграмма проекта.ч [?]

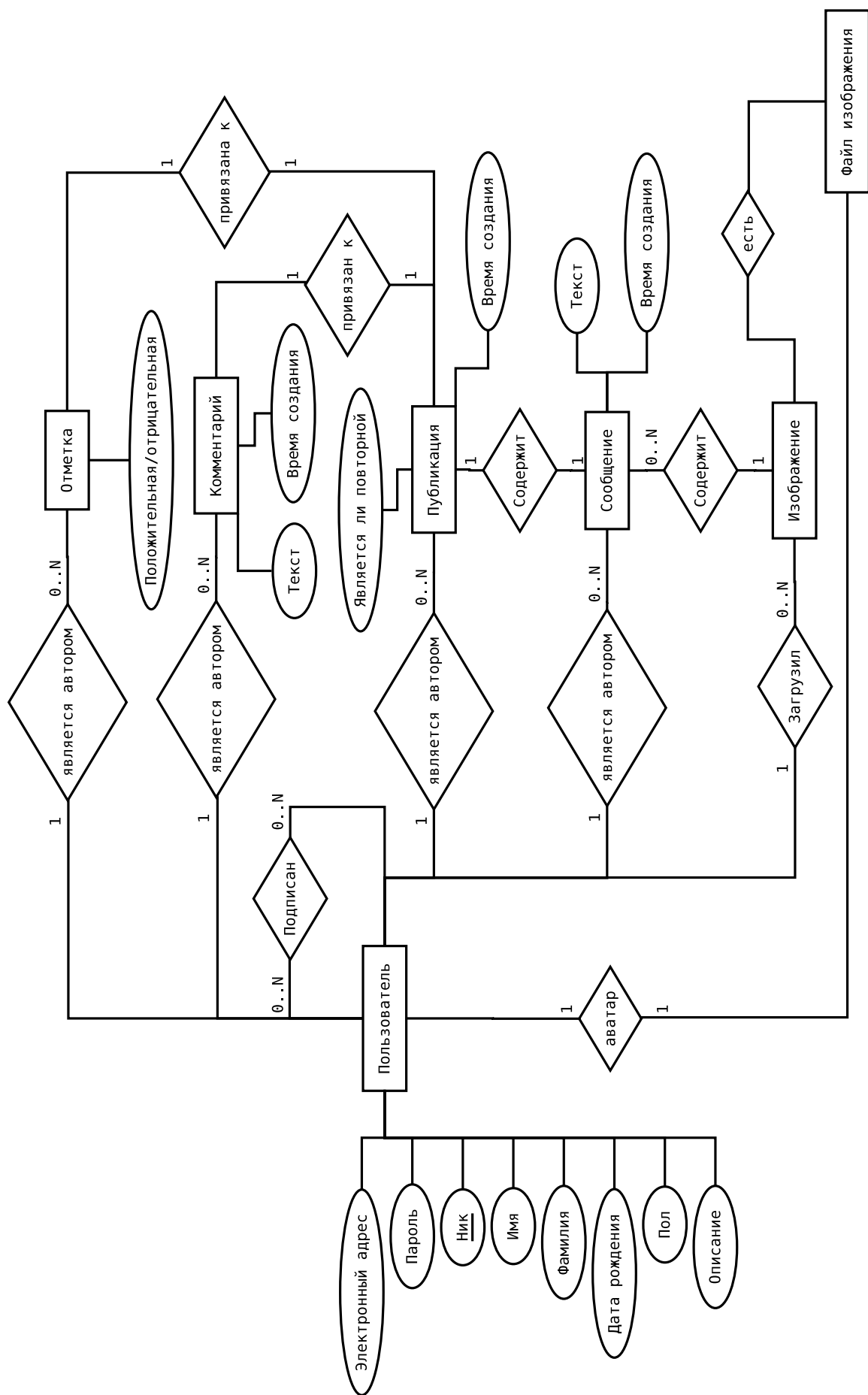


Рисунок 1.1 — ER-диаграмма

2 Конструкторский раздел

В данном разделе проектируется новая всячина.

2.1 Архитектура всячины

Проверка параграфа. Вроде работает.

Вторая проверка параграфа. Опять работает.

Вот.

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

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

Формула 2.1 совершенно бессмысленна.

$$a = cb \tag{2.1}$$

Окружение `cases` опять работает (см. 2.2), спасибо И. Короткову за исправления..

$$a = \begin{cases} 3x + 5y + z, & \text{если хорошо} \\ 7x - 2y + 4z, & \text{если плохо} \\ -6x + 3y + 2z, & \text{если совсем плохо} \end{cases} \tag{2.2}$$

2.2 Подсистема всякой ерунды

Культурная вставка dot-файлов через утилиту `dot2tex` (рис. 2.1).

2.2.1 Блок-схема всякой ерунды

Кстати о заголовках

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

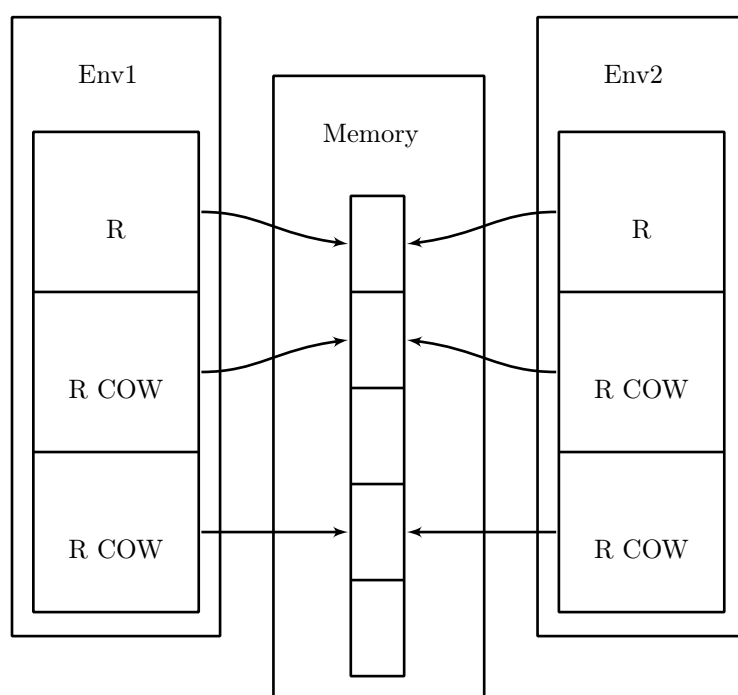


Рисунок 2.1 — Рисунок

3 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 3.1 — 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 3.2 — Грамматика

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

В листинге 3.3 работают русские буквы. Сильная магия. Однако, работает только во включаемых файлах, прямо в TeX нельзя.

Листинг 3.3 — Пример (`test.c`)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Комментарий на русском с пробелами */
6     printf("Это строка с пробелами и русскими буквами");
7
8     return 0;
9 }
```

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

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

4 Экспериментальный раздел

В данном разделе проводятся вычислительные эксперименты. А на рис. 4.1 показана схема мыслительного процесса автора...

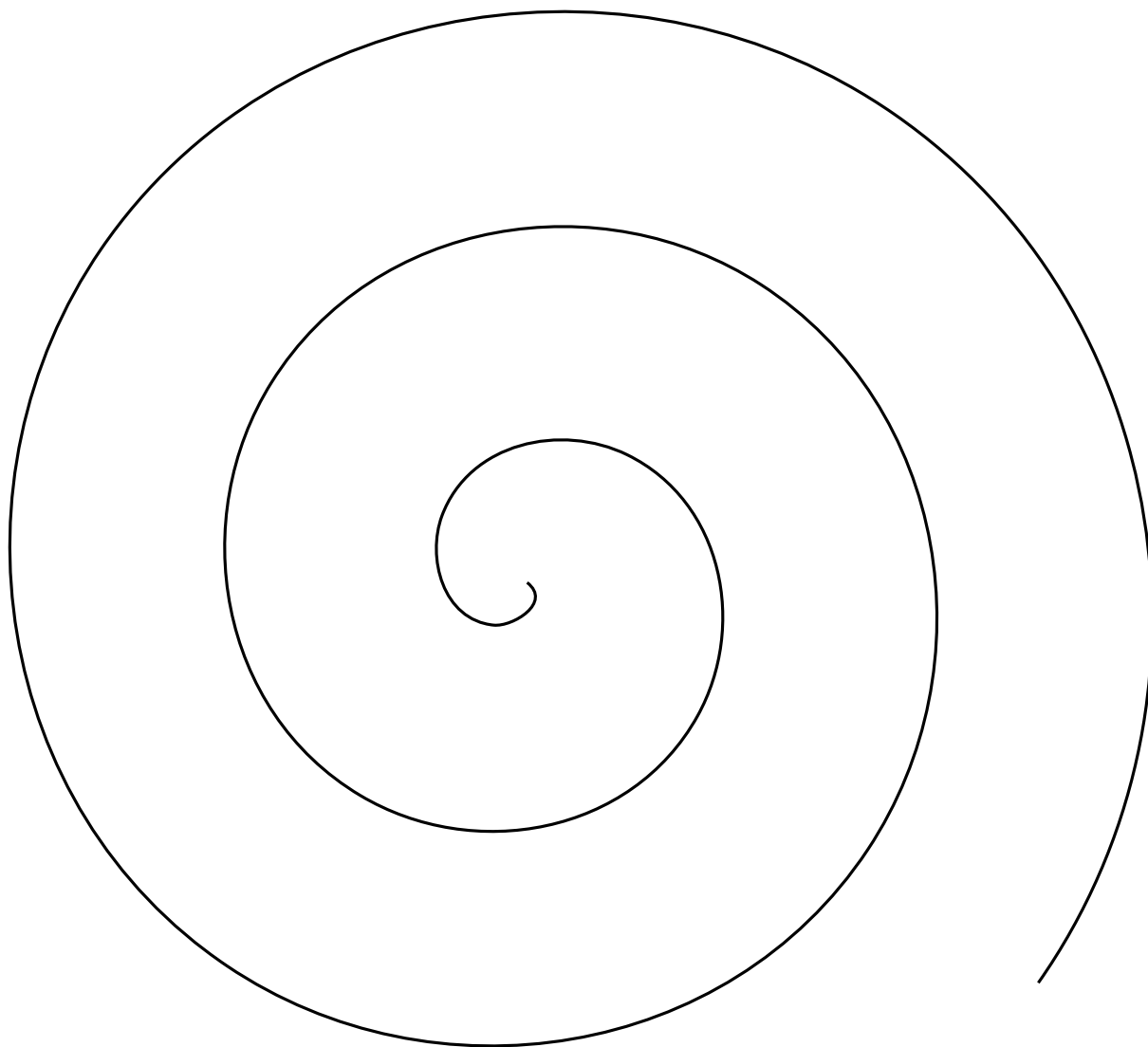


Рисунок 4.1 — Как страшно жить

Заключение

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

Список использованных источников