

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПЕТРА ВЕЛИКОГО»
ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
ВЫСШАЯ ШКОЛА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
И СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

**Отчет о прохождении производственной технологической
(проектно-технологической) практики
на тему: «Разработка системы учета рабочего времени для сотрудников»**

Родионова Ильи Алексеевича, гр. 3530903/90301

Направление подготовки: 09.03.03 Прикладная информатика.

Место прохождения практики: СПбПУ, ИКНТ, ВШИСиСТ.

Сроки практики: с 11.06.2022 по 09.07.2022.

Руководитель практики от ФГАОУ ВО «СПбПУ»: Туральчук Константин
Анатольевич, к. т. н., доцент ВШИСиСТ.

Консультант практики от ФГАОУ ВО «СПбПУ»: Пархоменко Владимир
Андреевич, ассистент ВШИСиСТ.

Руководитель практической подготовки от профильной организации:
Карпушин Василий Борисович, технический директор Rem&Coil.

Оценка: _____

Руководитель практической подготовки
от ФГАОУ ВО «СПбПУ»

Туральчук К.А

Консультант практической подготовки
от ФГАОУ ВО «СПбПУ»

Пархоменко В.А.

Руководитель практической подготовки
от профильной организации:

Карпушин В.Б.

Обучающийся

И.А. Родионов

Дата: 09.07.2022

СОДЕРЖАНИЕ

Введение	3
Глава 1. Описание общей архитектуры системы	4
1.1. Описание предложенного задания	4
1.2. Описание предполагаемой архитектуры	5
1.3. Резюме	8
Глава 2. Название второй главы: разработка метода, алгоритма, модели исследования.....	9
2.1. Название параграфа.....	9
2.2. Название параграфа	9
2.2.1. Название подпараграфа	10
2.3. Название параграфа.....	11
2.4. Выводы	17
Глава 3. Название третьей главы: разработка программного обеспечения..	18
3.1. Название параграфа.....	18
3.2. Название параграфа	18
3.3. Выводы	18
Глава 4. Название четвёртой главы. Апробация результатов исследования, а именно: метода, алгоритма, модели исследования	19
4.1. Название параграфа.....	19
4.2. Название параграфа	19
4.3. Выводы	19
Заключение	20
Список использованных источников.....	21
Приложение 1. Краткие инструкции по настройке издательской системы \LaTeX	24
Приложение 2. Некоторые дополнительные примеры	28

ВВЕДЕНИЕ

В качестве темы практической работы была предложена разработка системы учета рабочего времени для сотрудников *Rem&Coil*. Сотрудники технического отдела данной компании одновременно работают над несколькими проектами и переключаются с одного проекта на другой. На данный момент вычисление затраченного времени на каждый проект производится вручную в конце месяца. Поэтому была предложена идея разработать систему которая позволит централизовать и автоматизировать сбор необходимой информации.

Актуальность работы заключается в том, что данная система позволит производить анализ продуктивности коллектива технического отдела на основе собранных данных. Также, она освободит руководителя от необходимости самостоятельно собирать информацию и производить расчеты.

Цель работы разработать систему (которая включает в себя серверное и web приложения) учета рабочего времени для сотрудников.

Задачи работы:

- A. Изучить технологии, позволяющие разрабатывать серверные и web приложения.
- B. Разработать приложения, которые необходимы для корректной работы системы.
- C. Развернуть разработанную систему.
- D. Произвести ручное тестирование всего функционала.

Предполагаемые результаты:

- A. Обзор изученных технологий и библиотек;
- B. Разработка и внедрение системы;
- C. Результаты тестирования функционал.

ГЛАВА 1. ОПИСАНИЕ ОБЩЕЙ АРХИТЕКТУРЫ СИСТЕМЫ

Согласно «Чистой Архитектуре» Роберта Мартина [4], основная архитектура системы не должна зависеть от конкретных технологий (а скорее наоборот, технологии должны служить для реализации выбранной архитектуры), поэтому начать проектирование я решил именно с неё. Это позволит сразу определиться с форматом общения приложений между собой, а так же сформировать набор технологий которые позволят реализовать выбранную архитектуру.

1.1. Описание предложенного задания

Руководитель технического отдела сформулировал следующие критерии, которым должна соответствовать разработанная система:

- 1 Для взаимодействия с системой необходим веб-интерфейс, который должен быть адаптирован как для мобильных телефонов, так и для экрана монитора.
- 2 Поддерживаться две роли: администратор и обычный пользователь.
- 3 Админ создает проекты или архивирует их. На данном этапе для проектов храниться только название. Проекты доступны всем существующим пользователям (в том числе и администратору).
- 4 На странице проектов для каждого из них доступны следующие операции: пуск/пауза.
- 5 По нажатию на Пуск начинается отсчет времени. Если он уже был запущен ранее - он продолжается. Одновременно для одного пользователя может идти отсчет только по одному проекту. При активации проекта другой запущенный встает на паузу.
- 6 Должен быть экран с статистикой по текущему месяцу. Статистика должна отображаться как по проекту в целом, так и по пользователям отдельно. Пример текущего формата вывода статистики представлен на рис.1.1 и рис.1.2

Наименование	%	% кор
РЭМ	22	23
НЭМ	0	0
ПЭМ	0	0
ИС2000М	23	23
ГП4000	49	49
СФ2000М	5	5
	99	77

Рис.1.1. Пример текущего формата общей статистики

Василий							
Дата	День	РЭМ	НЭМ	ПЭМ	ИС2000М	ГП4000	СФ2000М
01.дек	Вт	1			1	6	8
02.дек	Ср	3			1	4	8
03.дек	Чт	2			3	3	8
04.дек	Пт	2			3	3	8
05.дек	Сб						0
06.дек	Вс						0
07.дек	Пн	1			1	6	8
08.дек	Вт	3			1	4	8
09.дек	Ср	2			3	3	8
10.дек	Чт	2			3	3	8
11.дек	Пт	1			3	4	8
12.дек	Сб						0
13.дек	Вс						0
14.дек	Пн	1			3	3	1
15.дек	Вт	3			2	2	1
16.дек	Ср	1			2	4	1
17.дек	Чт	4				4	8
18.дек	Пт	1			2	4	1
19.дек	Сб						0
20.дек	Вс						0
21.дек	Пн	1			1	5	1
22.дек	Вт				1	5	2
23.дек	Ср						0
24.дек	Чт						0
25.дек	Пт						0
26.дек	Сб						0
27.дек	Вс						0
28.дек	Пн						0
29.дек	Вт						0
30.дек	Ср						0
31.дек	Чт						0
ВСЕГО		28	0	0	30	63	7
		РЭМ	НЭМ	ПЭМ	ИС2000М	ГП4000	СФ2000М

Рис.1.2. Пример текущего формата общей статистики

1.2. Описание предполагаемой архитектуры

После анализа предложенного задания и исходя из предметной области, можно сделать вывод, что наиболее предпочтительной архитектурой системы является архитектура «Клиент — сервер».

Архитектура «Клиент — сервер» - сборное понятие, состоящее из двух взаимодополняющих компонентов: сервера и клиента.

Клиент — это программа, с которой работает пользователь. Он работает в браузере или с desktop-приложением. В разрабатываемой системе в качестве клиента будет использоваться веб-интерфейс.

Сервер - это компьютер, на котором хранится само приложение. Весь код, вся логика, все дополнительные материалы и справочники. Так же сервер обычно разделяют на две сущности: сервер с основной логикой приложения и сервер с базой данных.

Простейшая схема клиент-серверной архитектуры представлен на рисунке рис.1.3

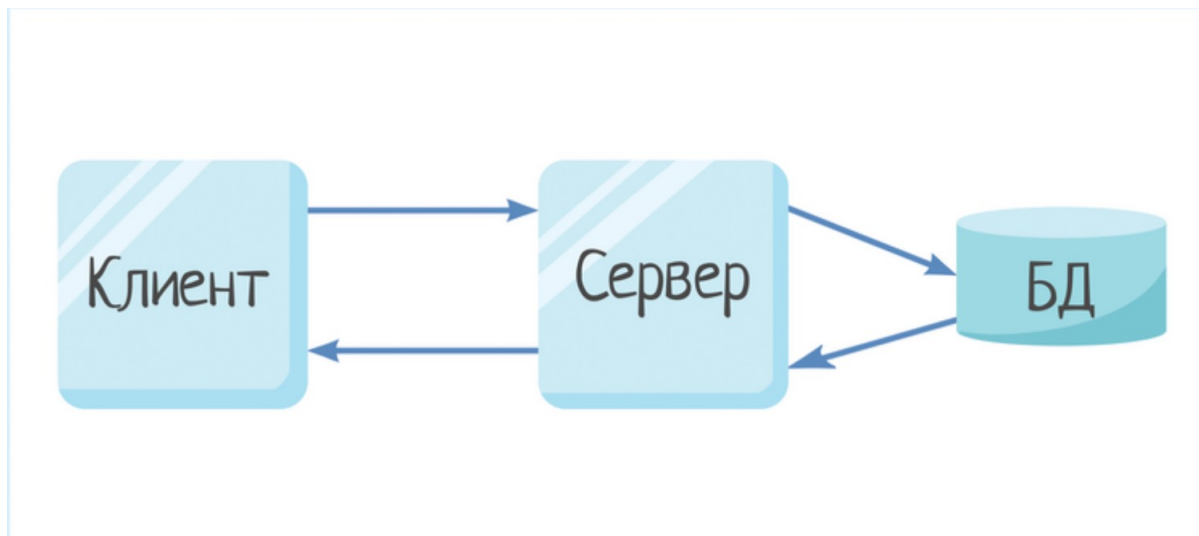


Рис.1.3. Архитектура «Клиент — сервер»

Основной причиной использования данной архитектуры является **централизация обработки входящих запросов**: запросы от всех клиентов обрабатываются и хранятся в одном месте, что позволяет упростить синхронизацию состояний между клиентами.

В клиент-серверной архитектуре клиент и сервер находятся на разных компьютерах и являются разными приложениями, поэтому им необходим интерфейс для общения. Наиболее популярным на данный момент *RESTful API*.

RESTful API[6] — это интерфейс, используемые двумя компьютерными системами для безопасного обмена информацией через Интернет. Он определяет правила, которым необходимо следовать для связи с другими программными системами. Разработчики внедряют или создают API-интерфейсы, чтобы другие приложения могли программно взаимодействовать с их приложениями.

Основной концепцией REST является то, что все данные представляются в виде ресурсов, которые хранятся в определенных местах. Клиент связывается с сервером с помощью API, когда ему требуется какой-либо ресурс.

Основные этапы запроса REST API:

- 1 Клиент отправляет запрос на сервер. Руководствуясь документацией API, клиент форматирует запрос таким образом, чтобы его понимал сервер.
- 2 Сервер аутентифицирует клиента и подтверждает, что клиент имеет право сделать этот запрос.

- 3 Сервер получает запрос и внутренне обрабатывает его.
- 4 Сервер возвращает ответ клиенту. Ответ содержит информацию, которая сообщает клиенту, был ли запрос успешным. Также запрос включает сведения, запрошенные клиентом.

Традиционно, для передачи запросов от клиента к серверу используется протокол передачи гипертекста (*HTTP*). В качестве адреса доступа к ресурсу используется URL. Также, протокол HTTP поддерживает большой набор методов обращения к серверу, наиболее используемые это GET, POST, PUT, DELETE. Метод HTTP сообщает серверу, что ему необходимо сделать с ресурсом. Например, если метод у запроса GET то серверу необходимо вернуть информацию о данном ресурсе, а если DELETE - то удалить данный ресурс.

После обработки пришедшего запроса сервер должен сформировать ответ. Каждый ответ сервера должен содержать код ответа, по которому клиент понимает результат выполнения запроса. Например, коды 2XX указывают на успешное выполнение, а коды 4XX и 5XX — на ошибки. Также большинство ответов сервера могут содержать тело ответа, в котором храниться ответ сервера, например определенное представление ресурса.

Так как клиент и сервер должны пересылать представления ресурсов, необходимо выбрать определенный формат этого представления. В качестве такого формата я выбрал JSON, ввиду его простоты и гибкости. На рис.1.4 представлен пример ресурса в формате JSON.

```
[
  {
    "Id": 0,
    "FirstName": "string",
    "LastName": "string",
    "Name": "string",
    "EmailAddress": "string",
    "TerritoryId": 0
  }
]
```

Рис.1.4. JSON

Я выбрал RESTful API как основной интерфейс обращения к серверу в силу следующих его преимуществ:

- Системы, реализующие REST API, могут эффективно масштабироваться благодаря оптимизации взаимодействия между сервером и клиентом по REST.

- Веб-службы RESTful поддерживают полное разделение клиента и сервера. Они упрощают и разделяют различные серверные компоненты, чтобы каждая часть могла развиваться независимо. Изменения платформы или технологии в серверном приложении не влияют на клиентское приложение. Возможность разделения функций приложения на уровни еще больше повышает гибкость. Например, разработчики могут вносить изменения в уровень базы данных, не переписывая логику приложения.
- REST API не зависит от используемой технологии. Вы можете создавать как клиентские, так и серверные приложения на разных языках программирования, не затрагивая структуру API. Также можно изменить базовую технологию на любой стороне, не влияя на обмен данными.

1.3. Резюме

После анализа поставленной задачи, я решил разбить систему на два приложения: клиент и сервер. В качестве протокола взаимодействия между ними был выбран протокол HTTP, а также в качестве интерфейса обращения к серверному приложению был выбран RESTful API.

Данная архитектура простая в реализации, а так же позволяет в будущем совершенствовать и масштабировать разработанную систему.

ГЛАВА 2. НАЗВАНИЕ ВТОРОЙ ГЛАВЫ: РАЗРАБОТКА МЕТОДА, АЛГОРИТМА, МОДЕЛИ ИССЛЕДОВАНИЯ

Глава посвящена более подробным примерам оформления текстово-графических объектов.

В параграфе 2.1 приведены примеры оформления многострочной формулы и одиночного рисунка. Параграф 2.2 раскрывает правила оформления перечислений и псевдокода. В параграфе 2.3 приведены примеры оформления сложносоставных рисунков, длинных таблиц, а также теоремоподобных окружений.

2.1. Название параграфа

Все формулы, размещенные в отдельных строках, подлежат нумерации, например, как формулы (2.1) и (2.2) из [20].

$$A^{\uparrow} = \{m \in M \mid gIm \forall g \in A\}; \quad (2.1)$$

$$B^{\downarrow} = \{g \in G \mid gIm \forall m \in B\}. \quad (2.2)$$

Обратим внимание, что формулы содержат знаки препинания и что они выровнены по левому краю (с помощью знака & окружения align).

На рис.2.1 приведёна фотография Нового научно-исследовательского корпуса СПбПУ.



Рис.2.1. Новый научно-исследовательский корпус СПбПУ [30]

2.2. Название параграфа

Название параграфа оформляется с помощью команды `\section{...}`, название главы — `\chapter{...}`.

2.2.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.

Использование подпараграфов в основной части крайне не рекомендуется. В случае использования, необходимо вынести данный номер в содержание. Название подпараграфа оформляется с помощью команды `\subsubsection{...}`.

Вместо подпараграфов рекомендовано использовать перечисления.

Перечисления могут быть с нумерационной частью и без неё и использоваться с иерархией и без иерархии. Нумерационная часть при этом формируется следующим способом:

1. в перечислениях *без иерархии* оформляется арабскими цифрами с точкой (или длинным тире).
2. В перечислениях *с иерархией* — в последовательности сначала прописных латинских букв с точкой, затем арабских цифр с точкой и далее — строчных латинских букв со скобкой.

Далее приведён пример перечислений с иерархией.

- A. Первый пункт.
- B. Второй пункт.
- C. Третий пункт.
- D. По ГОСТ 2.105–95 [2] первый уровень нумерации идёт буквами русского или латинского алфавитов (*для определенности выбираем английский алфавит*), а второй — цифрами.

1. В данном пункте лежит следующий нумерованный список:

- a) первый пункт;
- b) третий уровень нумерации не нормирован ГОСТ 2.105–95 (*для определенности выбираем английский алфавит*);
- c) обращаем внимание на строчность букв в этом нумерованном и следующем маркированном списке:
 - первый пункт маркированного списка.

E. Пятый пункт верхнего уровня перечисления.

Маркированный список (без нумерационной части) используется, если нет необходимости ссылки на определенное положение в списке:

- первый пункт с *маленькой буквы* по правилам русского языка;
- второй пункт с *маленькой буквы* по правилам русского языка.

Оформление псевдокода необходимо осуществлять с помощью пакета `algorithm2e` в окружении `algorithm`. Данное окружение интерпретируется в шаблоне как рисунок. Пример оформления псевдокода алгоритма приведён на рис.2.2.

Algorithm

```

Input: the many-valued context  $\mathbb{M} \stackrel{\text{def}}{=} (G, M, W, J)$ , the class membership
 $\varepsilon : G \rightarrow K$ 
Output: positive and negative binary contexts  $\overline{\mathbb{K}}_+ \stackrel{\text{def}}{=} (\overline{G}_+, M, I_+)$ ,
 $\overline{\mathbb{K}}_- \stackrel{\text{def}}{=} (\overline{G}_-, M, I_-)$  such that i-tests found in  $\overline{\mathbb{K}}_+$  are diagnostic tests
in  $M$ , and objects from  $\overline{\mathbb{K}}_-$  are counter-examples
1. for  $\forall g_i, g_j \in G$  do
2.   if  $i < j$  then
3.      $\overline{G} \leftarrow (g_i, g_j);$ 
4.   for  $\forall (g_i, g_j) \in \overline{G}$  do
5.     if  $m(g_i) = m(g_j)$  then
6.        $(g_i, g_j)Im;$ 
7.     if  $\varepsilon(g_i) = \varepsilon(g_j)$  then
8.        $\overline{G}_+ \leftarrow (g_i, g_j);$ 
9.     else  $\overline{G}_- \leftarrow (g_i, g_j);$ 
10.   $I_+ = I \cap (\overline{G}_+ \times M), I_- = I \cap (\overline{G}_- \times M);$ 
11.  for  $\forall \overline{g}_+ \in \overline{G}_+, \forall \overline{g}_- \in \overline{G}_-$  do
12.    if  $\overline{g}_+^\uparrow \subseteq \overline{g}_-^\uparrow$  then
13.       $\overline{G}_+ \leftarrow \overline{G}_+ \setminus \overline{g}_+;$ 

```

Рис.2.2. Псевдокод алгоритма `DiagnosticTestsScalingAndInferring` [28]

Обратим внимание, что можно сослаться на строчку 1 псевдокода из рис.2.2.

2.3. Название параграфа

Одиночные формулы также, как и отдельные формулы в составе группы, могут быть размещены в несколько строк. Чтобы выставить номер формулы напротив средней строки, используйте окружение `multlined` из пакета `mathtools`

следующим образом [20]:

$$\begin{aligned}
 (A_1, B_1) &\leq (A_2, B_2) \Leftrightarrow \\
 &\Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow \\
 &\Leftrightarrow B_2 \subseteq B_1.
 \end{aligned}
 \tag{2.3}$$

Используя команду `\labelcref{...}` из пакета `cleveref`, допустимо оформить ссылку на несколько формул, например, (2.1–2.3).

Пример оформления четырёх иллюстраций в одном текстово-графическом объекте приведён на рис.2.3. Это возможно благодаря использованию пакета `subcaption`.

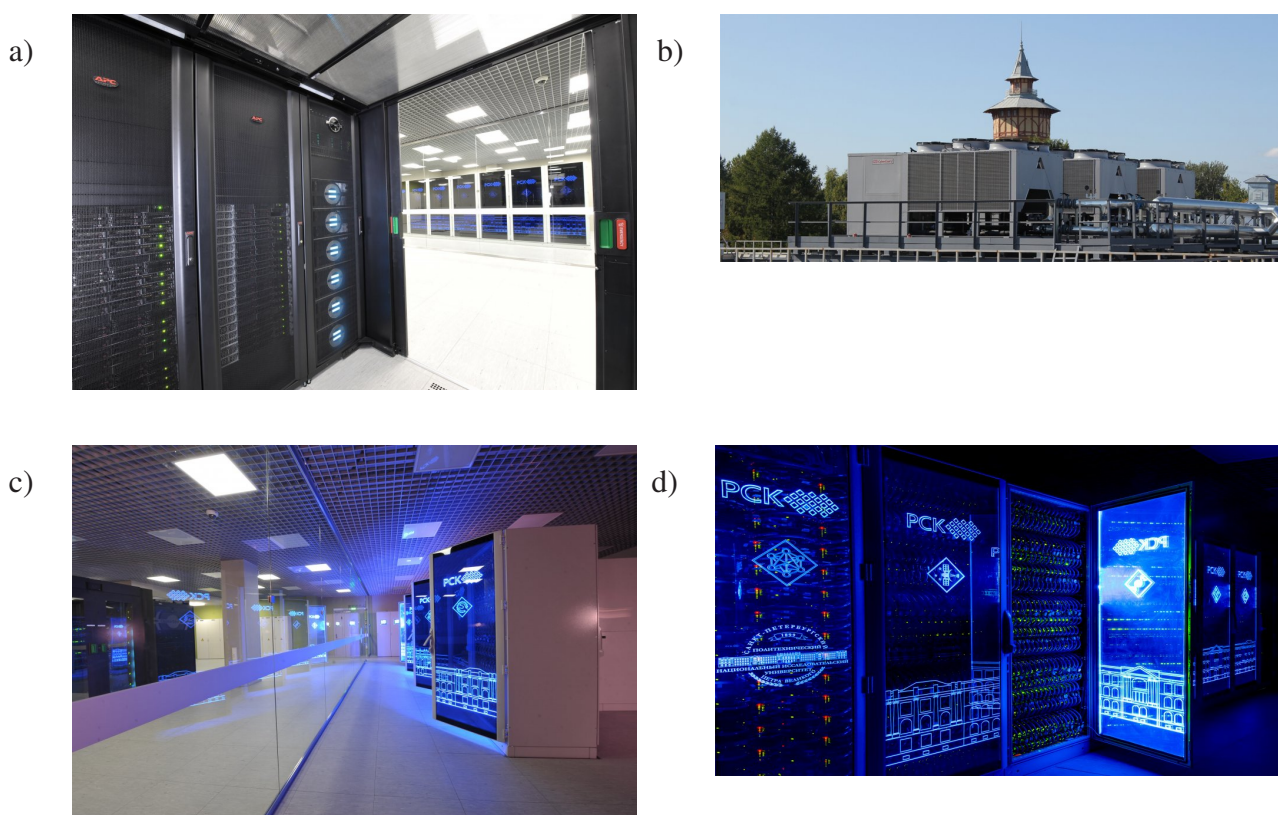


Рис.2.3. Фотографии суперкомпьютерного центра СПбПУ [30]: *a* — система хранения данных и узлы NUMA-вычислителя; *b* — холодильные машины на крыше научно-исследовательского корпуса; *c* — машинный зал; *d* — элементы вычислительных устройств

Далее можно ссылаться на составные части данного рисунка как на самостоятельные объекты: рис.2.3а, рис.2.3b, рис.2.3с, рис.2.3d или на три из четырёх изображений одновременно: рис.2.3а–2.3с.

Приведём пример табличного представления данных с записью продолжения на следующей странице на табл.2.1.

Таблица 2.1

Пример задания данных из [29] (с повтором для переноса таблицы на новую страницу)

G	m_1	m_2	m_3	m_4	K
1	2	3	4	5	6
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1

Продолжение табл. 2.1

1	2	3	4	5	6
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.2

Пример представления данных для сквозного примера по ВКР [29]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.3

Пример задания данных в табличном виде из [29] (с помощью окружения minipage)

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2



Рис.2.4. Новый научно-исследовательский корпус СПбПУ [30] (с помощью окружения minipage)

Вопросы форматирования текстово-графических объектов (окружений) не регламентированы в известных нам ГОСТах, поэтому предлагаем придерживаться следующих правил:

- **полужирный текст** рекомендуем использовать только для названий стандартных окружений с нумерационной частью, например, для представления *впервые*: **определение 1.1, теорема 2.2, пример 2.3, лемма 4.5**;
- *курсив* рекомендуем использовать только для выделения переменных в формулах, служебной информации об авторах главы (статьи), важных терминов, представляемых по тексту, а также для всего тела окружений, связанных с получением *новых существенных результатов и их доказательством*: теорема, лемма, следствие, утверждение и другие.

По аналогии с нумерацией формул, рисунков и таблиц нумеруются и иные текстово-графические объекты, то есть включаем в нумерацию номер главы, например: теорема 3.1. для первой теоремы третьей главы монографии. Команды \LaTeX выставляют нумерацию и форматирование автоматически. Полный перечень команд для подготовки текстово-графических и иных объектов находится в подробных методических рекомендациях [8].

Для удобства авторов названия стандартных окружений, рекомендованных к использованию, приведены в табл.2.4, а в табл.2.5 перечислены имена специально разработанных окружений для шаблонов SPbPU.

На базе пакета `tikz` разработано большое количество расширений [17], например, `tikzcd`, которые мы рекомендуем использовать для оформления иллюстраций.

В случае, если авторам потребовалось новое окружение, то создать его можно в файле `my_folder/my_settings.tex` согласно правилам, приведённым ниже.

1. Для перехода в режим создания окружений следует указать:
 - `\theoremstyle{myplain}` — окружения с доказательствами или аксиомами
 - `\theoremstyle{mydefinition}` — окружения, не связанные с доказательствами или аксиомами.
2. В команде создания окружения следует ввести краткий псевдоним (`m-new-env`) и отображаемое в pdf имя окружения (Название_окружения):
 - `\newtheorem{m-new-env-second}{Название_окружения} - [chapter]`.

Таблица 2.4

Стандартные окружения

Название окружения	Назначение
center	центрирование, аналог команды <code>\centering</code> , но с добавлением нежелательного пробела, поэтому лучше избегать применения <code>center</code>
itemize	перечисления, в которых нет необходимости нумеровать пункты (немаркированные списки)
enumerate	перечисления с нумерацией (немаркированные списки)
refsection	создание отдельных библиографических списков для глав
tabular	оформление таблиц
table	автоматическое перемещение по тексту таблиц, оформленных, например, с помощью <code>tabular</code> , для минимизации пустых пространств
longtable	оформление многостраничных таблиц
tikzpicture	создание иллюстраций с помощью пакета <code>tikz</code> [17]
figure	автоматическое перемещение по тексту рисунков, оформленных например, с помощью <code>tikz</code> или подключенных с помощью команды <code>\includegraphics</code> , для минимизации пустых пространств
subfigure	оформление вложенных рисунков в составе <code>figure</code>
algorithm	оформление псевдокода на основе пакета <code>algorithm2e</code> [18]
minipage	оформление рисунков и таблиц без функций автоматического перемещения по тексту для минимизации пустых пространств
equation	оформление выключенных (не встроенных в текст с помощью <code>\$...\$</code>) одиночных формул на одной строке
multilined	оформление выключенных (не встроенных в текст с помощью <code>\$...\$</code>) одиночных формул в несколько строк
aligned	оформление нескольких формул с выравниванием по символу <code>&</code> .

Таблица 2.5

Специальные окружения

Название окружения	Текстово-графический объект
abstr	реферат (abstract)
m-theorem	теорема
m-corollary	следствие
m-proposition	утверждение
m-lemma	лемма
m-axiom	аксиома
m-example	пример
m-definition	определение
m-condition	условие
m-problem	проблема
m-exercise	упражнение
m-question	вопрос
m-hypothesis	гипотеза

Теорема 2.1 (о чем-то конкретном). *Текст теоремы полностью выделен курсивом. Допустимо математические символы не выделять курсивом, если это искажает их значения. Используется абзацный отступ, так как “Абзацы в тексте начинают отступом” в соответствии с ГОСТ 2.105–95. Название теоремы допустимо убрать. Доказательство окончено.*

Доказательство теоремы 2.1, леммы, утверждений, следствий и других подобных окружений (в последнем абзаце) завершаем предложением в котором сказано, что доказательство окончено. Например, доказательство теоремы 2.1 окончено.

Тело доказательства не выделяется курсивом. Тело следующих окружений также не выделяется сплошным курсивом: определение, условие, проблема, пример, упражнение, вопрос, гипотеза и другие.

Определение 2.1 (термин). В тексте определения только *важные термины* выделяются курсивом. Если определение носит лишь вспомогательный характер, то допустимо не использовать окружение `m-definition`, представляя текст определения в обычном абзаце. Ключевые термины при этом обязательно выделяются курсивом.

Вместо теоремо-подобных окружений для вставки небольших текстово-графических объектов иногда используются команды. Типичным примером такого подхода является команда `\footnote{text}`¹, где в аргументе `text` указывают текст *подстрочной ссылки (сноски)*. В них *нельзя добавлять веб-ссылки или цитировать литературу*. Для этих целей используется список литературы. Нумерация сносок сквозная по ВКР без точки на конце выставляется в шаблоне автоматически, однако в каждом приложении к ВКР нумерация, зависящая от номера приложения, выставляется префикс «П», например «П1.1» — первая сноска первого приложения.

2.4. Выводы

Текст заключения ко второй главе. Пример ссылок [7; 9—11; 14—16; 19; 21—23; 25; 26; 32], а также ссылок с указанием страниц, на котором отображены те или иные текстово-графические объекты [28, с. 96] или в виде мультицитаты

¹Внимание! Команда вставляется непосредственно после слова, куда вставляется сноска (без пробела). Лишние пробелы также не указываются внутри команды перед и после фигурных скобок.

на несколько источников [28, с. 96; 20, с. 46]. Часть библиографических записей носит иллюстративный характер и не имеет отношения к реальной литературе.

Короткое имя каждого библиографического источника содержится в специальном файле `my_biblio.bib`, расположенном в папке `my_folder`. Там же находятся исходные данные, которые с помощью программы `Viber` и стилевого файла `Biblatex-GOST` [12] приведены в списке использованных источников согласно ГОСТ 7.0.5-2008. Многообразные реальные примеры исходных библиографических данных можно посмотреть по ссылке [13].

Как правило, ВКР должна состоять из четырех глав. Оставшиеся главы можно создать по образцу первых двух и подключить с помощью команды `\input` к исходному коду ВКР. Далее в приложении 1 приведены краткие инструкции запуска исходного кода ВКР [27; 31].

В приложении 2 приведено подключение некоторых текстово-графических объектов. Они оформляются по приведенным ранее правилам. В качестве номера структурного элемента вместо номера главы используется «П» с номером главы. Текстово-графические объекты из приложений не учитываются в реферате.

ГЛАВА 3. НАЗВАНИЕ ТРЕТЬЕЙ ГЛАВЫ: РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

3.1. Название параграфа

3.2. Название параграфа

3.3. Выводы

Текст выводов по главе 3.

ГЛАВА 4. НАЗВАНИЕ ЧЕТВЁРТОЙ ГЛАВЫ. АПРОБАЦИЯ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ, А ИМЕННО: МЕТОДА, АЛГОРИТМА, МОДЕЛИ ИССЛЕДОВАНИЯ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

4.1. Название параграфа

4.2. Название параграфа

Пример ссылки на литературу [1; 3; 5; 24].

4.3. Выводы

Текст выводов по главе 4.

ЗАКЛЮЧЕНИЕ

Заключение (2 – 5 страниц) обязательно содержит выводы по теме работы, *конкретные предложения и рекомендации* по исследуемым вопросам. Количество общих выводов должно вытекать из количества задач, сформулированных во введении выпускной квалификационной работы.

Предложения и рекомендации должны быть органически увязаны с выводами и направлены на улучшение функционирования исследуемого объекта. При разработке предложений и рекомендаций обращается внимание на их обоснованность, реальность и практическую приемлемость.

Заключение не должно содержать новой информации, положений, выводов и т. д., которые до этого не рассматривались в выпускной квалификационной работе. Рекомендуются писать заключение в виде тезисов.

Последним абзацем в заключении можно выразить благодарность всем людям, которые помогали автору в написании ВКР.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Автономова Н. С.* Философский язык Жака Деррида. — М.: Российская политическая энциклопедия (РОССПЭН), 2011. — 510 с. — (Сер.: Российские Пропилеи).
2. Единая система конструкторской документации. Общие требования к текстовым документам [Текст]: ГОСТ 2.105–95. — Взамен ГОСТ 2.105—79, ГОСТ 2.906—71 ; введ. 1996—07—01. — Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2002. — 31 с. — (Сер.: Межгосударственный стандарт).
3. *Котельников И. А., Чеботаев П. З.* LaTeX по-русски. — 3-е изд. — Новосибирск: Сибирский Хронограф, 2004. — 496 с. — URL: <http://www.tex.uniya.ru/doc/kotelnikovchebotaev2004b.pdf> (дата обращения: 06.03.2019).
4. *Мартин Р.* Чистая архитектура. — Питер, 2021. — 352 с.
5. *Песков Н. В.* Поиск информативных фрагментов описаний объектов в задачах распознавания: дис. . . . канд. канд. физ.-мат. наук: 05.13.17 / Песков Николай Владимирович. — М., 2004. — 102 с.
6. Что такое RESTful API? — URL: <https://aws.amazon.com/ru/what-is/restful-api/> (дата обращения: 28.06.2022).
7. *Adams P.* The title of the work // The name of the journal. — 1993. — Vol. 4, no. 2. — P. 201–213.
8. Author and editor guide to prepare and submit the academic SPbPU editions to Clarivate Analytics: Book Citation Index Web of Science / V. Parkhomenko [et al.]. — 2018. — URL: https://github.com/ParkhomenkoV/SPbPU-BCI-template/blob/master/Author_guide_SPbPU-BCI.pdf (visited on 06.03.2019).
9. *Babington P.* The title of the work. Vol. 4. — 3rd ed. — The address: The name of the publisher, 1993. — 255 p. — (Ser.: 10).
10. *Badiou A.* Briefings on Existence: A Short Treatise on Transitory Ontology / ed. and trans. from the French, with an introd., by N. Madarasz. — NY: SUNY Press, 2006. — 190 p. — URL: https://books.google.ru/books?id=7HNkAT%5C_NFksC (visited on 05.12.2017).
11. *Caxton P.* The title of the work. — The address of the publisher, 1993. — 255 p.
12. *Domanov O.* BibLATEX support for GOST standard bibliographies. — URL: <https://ctan.org/pkg/biblatex-gost> (visited on 06.03.2019).

13. *Domanov O.* Biblatex-GOST examples. — URL: <http://ctan.altspu.ru/macros/latex/contrib/biblatex-contrib/biblatex-gost/doc/biblatex-gost-examples.pdf> (visited on 06.03.2019).
14. *Draper P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — The organization. The address of the publisher: The publisher, 1993. — (Ser.: 5).
15. *Eston P.* The title of the work // Book title. Vol. 4. — 3rd ed. — The address of the publisher: The name of the publisher, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).
16. *Farindon P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — 3rd ed. — The address of the publisher: The name of the publisher, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).
17. *Feuersanger C., Tantau T.* The TikZ and PGF packages. — URL: <https://ctan.org/pkg/pgf> (visited on 06.03.2019).
18. *Fiorio C.* The algorithm2e package. — URL: <https://ctan.org/pkg/algorithm2e> (visited on 06.03.2019).
19. *Gainsford P.* The title of the work / The organization. — 3rd ed. — The address of the publisher, 1993. — 255 p.
20. *Ganter B., Wille R.* Formal concept analysis: mathematical foundations. — Springer, Berlin, 1999. — 284 p.
21. *Harwood P.* The title of the work: Master's thesis / Harwood Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.
22. *Isley P.* The title of the work. — 1993.
23. *Joslin P.* The title of the work: diss. ... PhD in Engineering / Joslin Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.
24. *Kotelnikov I. A., Chebotaev P. Z.* LaTeX in Russian. — 3rd ed. — Novosibirsk: Sibiskiy Hronograph, 2004. — 496 p. — URL: <http://www.tex.uniyar.ac.ru/doc/kotelnikovchebotaev2004b.pdf> (visited on 06.03.2019); (in Russian).
25. *Lambert P.* The title of the work: tech. rep. / The institution that published. — The address of the publisher, 1993. — 255 p. — No. 2.
26. *Marcheford P.* The title of the work. — 1993.
27. MiKTeX web site. — URL: <https://miktex.org/> (visited on 06.03.2019).
28. Notes on relation between symbolic classifiers / X. Naidenova [et al.] // CEUR Workshop Proceedings / ed. by K. S. Watson B.W. — 2017. — Vol. 1921. — P. 88–103. — URL: <http://ceur-ws.org/Vol-1921/paper9.pdf> (visited on 19.12.2017).

29. *Peskov N. V.* Searching for informative fragments of object descriptions in the recognition tasks: diss. ... cand. phys.-math. sci.: 05.13.17 / Peskov Nickolay Vladimirovich. — M., 2004. — 102 p. — (in Russian).

30. SPbPU photo gallery. — URL: <http://www.spbstu.ru/media/photo-gallery/> (visited on 06.03.2019).

31. TeXstudio web site. — URL: <https://www.texstudio.org/> (visited on 06.03.2019).

32. The title of the work. Vol. 4 / ed. by P. Kidwelly. — The organization. The address of the publisher: The name of the publisher, 1993. — 255 p. — (Ser.: 5).

Приложение 1

Краткие инструкции по настройке издательской системы L^AT_EX

В SPbPU-BCI-template автоматически выставляются необходимые настройки и в исходном тексте шаблона приведены примеры оформления текстово-графических объектов, поэтому авторам достаточно заполнить имеющийся шаблон текстом главы (статьи), не вдаваясь в детали оформления, описанные далее. Возможный «быстрый старт» оформления главы (статьи) под Windows следующий^{П1.1}:

- A. Установка полной версии MikTeX [27]. В процессе установки лучше выставить параметр доустановки пакетов «на лету».
- B. Установка TexStudio [31].
- C. Запуск TexStudio и компиляция `my_chapter.tex` с помощью команды «Build&View» (например, с помощью двойной зелёной стрелки в верхней панели). Иногда, для достижения нужного результата необходимо несколько раз скомпилировать документ.
- D. В случае, если не отобразилась библиография, можно
 - воспользоваться командой Tools → Commands → Biber, затем запустив Build&View;
 - настроить автоматическое включение библиографии в настройках Options → Configure TexStudio → Build → Build&View (оставить по умолчанию, если сборка происходит слишком долго): `txs:///pdflatex | txs:///biber | txs:///pdflatex | txs:///pdflatex | txs:///view-pdf`.

В случае возникновения ошибок, попробуйте скомпилировать документ до последних действий или внимательно ознакомьтесь с описанием проблемы в log-файле. Бывает полезным переход (по подсказке TexStudio) в нужную строку в pdf-файле или запрос с текстом ошибки в поисковиках. Наиболее вероятной проблемой при первой компиляции может быть отсутствие какого-либо установленного пакета L^AT_EX.

В случае корректной работы настройки «установка на лету» все дополнительные пакеты будут скачиваться и устанавливаться в автоматическом режиме. Если доустановка пакетов осуществляется медленно (несколько пакетов за один запуск

^{П1.1} Вниманию! Пример оформления подстрочной ссылки (сноски).

компилятора), то можно попробовать установить их в ручном режиме следующим образом:

1. Запустите программу: меню → все программы → MikTeX → Maintenance (Admin) → MiKTeX Package Manager (Admin).
2. Пользуясь поиском, убедитесь, что нужный пакет присутствует, но не установлен (если пакет отсутствует воспользуйтесь сначала MiKTeX Update (Admin)).
3. Выделив строку с пакетом (возможно выбрать несколько или вообще все неустановленные пакеты), выполните установку Tools → Install или с помощью контекстного меню.
4. После завершения установки запустите программу MiKTeX Settings (Admin).
5. Обновите базу данных имен файлов Refresh FNDB.

Для проверки текста статьи на русском языке полезно также воспользоваться настройками Options → Configure TexStudio → Language Checking → Default Language. Если русский язык «ru_RU» не будет доступен в меню выбора, то необходимо вначале выполнить Import Dictionary, скачав из интернета любой русскоязычный словарь.

Далее приведены формулы (П1.2), (П1.1), рис.П1.2, рис.П1.1, табл.П1.2, табл.П1.1.

$$\pi \approx 3,141. \quad (\text{П1.1})$$



Рис.П1.1. Вид на гидробашню СПбПУ [30]

Представление данных для сквозного примера по ВКР [29]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

П1.1. Параграф приложения

П1.1.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.
Использование подподпараграфов в основной части крайне не рекомендуется.

П1.1.1.1. Название подподпараграфа

$$\pi \approx 3,141. \quad (\text{П1.2})$$



Рис.П1.2. Вид на гидробашню СПбПУ [30]

Представление данных для сквозного примера по ВКР [29]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Приложение 2

Некоторые дополнительные примеры

В приложении^{П2.1} приведены формулы (П2.2), (П2.1), рис.П2.2, рис.П2.1, табл.П2.2, табл.П2.1

$$\pi \approx 3,141.$$

(П2.1)



Рис.П2.1. Вид на гидробашню СПбПУ [30]

Таблица П2.1

Представление данных для сквозного примера по ВКР [29]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

^{П2.1}Внимание! Пример оформления подстрочной ссылки (сноски).

П2.1. Подраздел приложения

$$\pi \approx 3,141.$$

(П2.2)



Рис.П2.2. Вид на гидробашню СПбПУ [30]

Таблица П2.2

Представление данных для сквозного примера по ВКР [29]

<i>G</i>	<i>m</i> ₁	<i>m</i> ₂	<i>m</i> ₃	<i>m</i> ₄	<i>K</i>
<i>g</i> ₁	0	1	1	0	1
<i>g</i> ₂	1	2	0	1	1
<i>g</i> ₃	0	1	0	1	1
<i>g</i> ₄	1	2	1	0	2
<i>g</i> ₅	1	1	0	1	2
<i>g</i> ₆	1	1	1	2	2