



**Факультет Кибернетики и Информационной  
безопасности**

**Кафедра кибернетики (№ 22)**

Направление подготовки 09.03.04 Программная инженерия

**Расширенное содержание пояснительной записки**

к учебно-исследовательской работе студента на тему:

---

---

Группа

---

Студент

---

(подпись)

---

(ФИО)

Руководитель

---

(подпись)

---

(ФИО)

Научный консультант

---

(подпись)

---

(ФИО)

Оценка руководителя \_\_\_\_\_

Оценка комиссии \_\_\_\_\_

Члены комиссии

---

---

---

---

---

---

---

---

**Москва 2016**



Факультет кибернетики и информационной безопасности

КАФЕДРА КИБЕРНЕТИКИ

## Задание на УИР

Студенту гр. Б. - \_\_\_\_\_  
(группа) (фио)

### ТЕМА УИР

### ЗАДАНИЕ

№ п/п	Содержание работы	Форма отчетности	Срок исполнения	Отметка о выполнении Дата, подпись
1.	<b>Аналитическая часть</b> (указываются предмет и цели анализа)			
1.1.	Изучение и сравнительный анализ ... с целью...			
1.2.	Изучение и анализ ... для...			
1.3.	Анализ ... применительно к задачам...			
1.4.	Анализ возможностей... (для..., применительно к..., и т.п.)			
1.5.	Оформление расширенного содержания пояснительной записки (РСПЗ)	Текст РСПЗ	20.10.2016	
2.	<b>Теоретическая часть</b> (указываются используемые и разрабатываемые модели, методы, алгоритмы)			
2.1.	Используется ... (модель, метод, алгоритм(ы)...)	Модель/ алгоритм/метод...		
2.2.	Выбор/разработка...			
2.3.	Разработка...			
2.4.	Модификация... (алгоритма, модели, и т.п.) для ... Адаптация ... для...			
3.	<b>Инженерная часть</b> (указывается, что конкретно необходимо спроектировать, а также используемые для этого методы, технологии и инструментальные средства)			
3.1.	Проектирование ... (системы, подсистемы, модуля...)			
3.2.	Использовать методологию проектирования....			
3.3.	Разработать архитектуру для... (с учетом требований к...)			

3.4.	Результаты проектирования оформить с помощью.... При проектировании использовать язык... (например, IDEF, или UML)			
4.	<b>Технологическая и практическая часть</b> (указывается, что конкретно должно быть реализовано и протестировано, а также используемые для этого методы, инструментальные средства, технологии)			
4.1.	Реализовать... (систему, подсистему, модуль...)	Исполняемые файлы, исходный текст		
4.2.	Протестировать... с помощью... Разработать тестовые примеры для...	Исполняемые файлы, исходные тексты тестов и тестовых примеров		
4.3.	Реализация должна иметь форму/обладать качествами...			
4.4.	Ожидаемым результатом является программная система/программный комплекс/программное обеспечение... со следующими отличительными характеристиками...			
4.5.	При реализации использовать технологию/платформу...			
5.	Оформление пояснительной записки (ПЗ) и иллюстративного материала для доклада.	Текст ПЗ, презентация	15.12.2016	

## ЛИТЕРАТУРА

[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	
[8]	
[9]	
[10]	

Дата выдачи задания:

Руководитель \_\_\_\_\_

(ФИО)

«    »                      2016г.

Студент \_\_\_\_\_

(ФИО)

## Реферат

Пояснительная записка содержит XXX страниц (из них XX страниц приложений). Количество использованных источников – XX. Количество приложений – X.

Ключевые слова: ....

Целью данной работы является ...

В первой главе проводится обзор и анализ ...

Во второй главе описываются использованные и разработанные/модифицированные методы-/модели/алгоритмы ....

В третьей главе приводится описание программной реализации и экспериментальной проверки ....

В приложении А описаны основные требования к форматированию пояснительных записок к дипломам и (магистерским) диссертациям.

В приложении В представлена общая структура пояснительной записки.

В приложении С приведены некоторые дополнительные комментарии к использованию данного шаблона.

# Содержание

<b>Введение</b>	<b>7</b>
<b>1 Анализ проблематики ...</b>	<b>8</b>
1.1 Общий анализ проблематики ...	8
1.2 Анализ особенностей ...	8
1.3 Сравнительный анализ алгоритмов ...	8
1.4 Сравнительный анализ программных средств ...	8
1.5 Выводы ...	10
1.6 Постановка задачи дипломной работы/курсового проекта ...	10
<b>2 Разработка моделей и алгоритмов ...</b>	<b>11</b>
2.1 Модель системы ...	11
2.2 Метод решения задачи для ...	11
2.3 Алгоритмы вычисления ...	11
2.4 Обобщенная архитектура и интерфейсы ...	11
2.5 Выводы ...	11
<b>3 Результаты проектирования ...</b>	<b>13</b>
3.1 Использование методики «такой-то» для проектирования программных систем «такого-то типа» ...	13
3.2 Общая архитектура системы ...	13
3.3 Архитектура подсистемы 1... ..	13
3.4 Архитектура подсистемы $N$ ... ..	13
3.5 Проектирование протокола взаимодействия подсистем $X$ и $Y$ ...	13
3.6 Выводы ...	13
<b>4 Реализация и экспериментальная проверка ...</b>	<b>14</b>
4.1 Выбор инструментальных средств ...	14
4.2 Состав и структура реализованного программного обеспечения ...	14
4.3 Основные сценарии работы пользователя ...	15
4.4 Разработка тестовых примеров ...	15

4.5	Сравнение реализованного программного обеспечения с существующими аналогами	16
4.6	Выводы . . . . .	16
<b>Заключение</b>		<b>17</b>
<b>Приложения</b>		<b>19</b>
<b>A</b>	<b>Основные правила форматирования</b>	<b>19</b>
<b>B</b>	<b>Общая структура пояснительной записки</b>	<b>21</b>
<b>C</b>	<b>Правила использования шаблона</b>	<b>23</b>

## Введение

Введение всегда содержит краткую характеристику работы по следующим аспектам:

- актуальность:
  - кто и почему в настоящее время интересуется данной проблематикой (в т.ч. для решения каких задач могут быть полезны исследования в данной области),
  - краткая история вопроса (в формате год-фамилия-что сделал),
  - нерешенные вопросы/проблемы;
- новизна работы (что нового привносится данной работой);
- оригинальная суть исследования;
- содержание по главам (по одному абзацу на главу).

Общий объем введения должен не превышать 1,5 страниц (для ПЗ к УИРам может быть чуть меньше).

# 1. Анализ проблематики ...

Это обзорно-аналитическая глава, в которой требуется отразить:

- результат изучения различных существующих методов решения задач в рамках проблематики УИРа/диплома (иногда даже в смежных областях), это обзорный аспект, который пишется, в основном, на основе имеющейся литературы или/и программного обеспечения;
- сравнение (с какой-либо определенной целью) этих методов и средств.

Приведенные ниже названия пунктов являются очень примерными, их состав и структура сильно зависят от специфики конкретной работы.

## 1.1 Общий анализ проблематики ...

*Аннотация.* В случае РСПЗ, так оформляется аннотация к разделу. Такая же аннотация, только более общая, должна быть для главы. После аннотации может следовать рабочая или финальная версия текста соответствующего раздела. В случае ПЗ, таких аннотаций быть не должно.

...

Результаты анализа полезно оформлять в виде таблиц (см. табл. 1.1).

## 1.2 Анализ особенностей ...

Сначала приведем пример более сложной таблицы (см. табл. 1.2 и 1.2).

А теперь, продемонстрируем, как должна выглядеть иллюстрация (см. рис. 1.1).

...

## 1.3 Сравнительный анализ алгоритмов ...

...

## 1.4 Сравнительный анализ программных средств ...

...

Таблица 1.1 – Результаты сравнения нескольких программных систем

№	Название системы	Показатель 1	Показатель 2	Показатель 3	Показатель 4
---	------------------	--------------	--------------	--------------	--------------



Таблица 1.2 – Таблица с длинным, многострочным названием, чтобы показать, как форматируется такой заголовок

№	Название системы	Показатель 1	Показатель 2	Показатель 3	Показатель 4
---	------------------	--------------	--------------	--------------	--------------

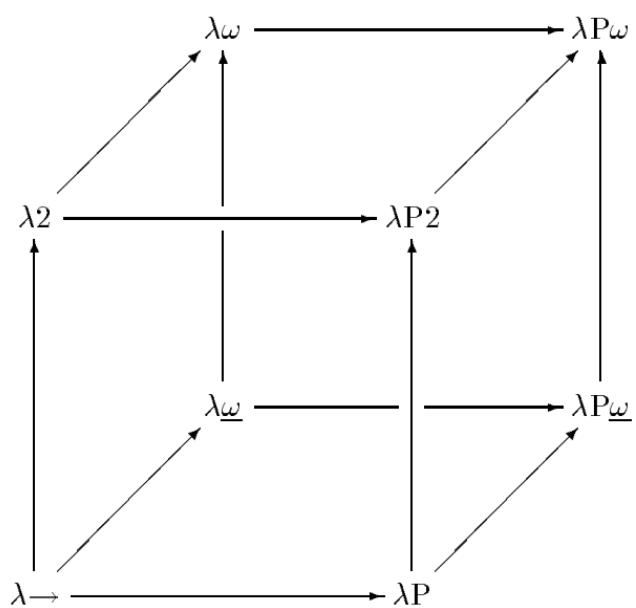


Рис. 1.1 – Ламбда-куб Барендрегта

## **1.5 Выводы**

Тут пишем выводы по результатам анализа: что и с какой целью было проанализировано, какие выводы из этого сделаны, как они повлияли (должны повлиять) на дальнейший ход работы. Результаты анализа приводятся по пунктно, основные выводы из проделанного анализа. Например:

1. Выполнен сравнительный анализ таких-то формальных систем с точки зрения применимости к решению такой-то задачи. Ни одна из проанализированных напрямую не подходит, поэтому требуется разработать вариацию на основе системы такой-то.
2. Были проанализированы варианты программных архитектур на основе систем. С учетом требований к поддержке больших объемов данных и высоких требований к потенциалу модернизруемости, была выбрана за основу такая-то архитектура.
3. Сравнительный анализ таких-то библиотек показал, что библиотека X проще в использовании, но менее производительна, в то время как библиотека Y обеспечивает высокую производительность, но и требует значительных трудозатрат для использования. В связи с такими-то соображениями были принято решение использовать такую-то библиотеку.

## **1.6 Постановка задачи дипломной работы/курсового проекта**

Это всегда последний пункт. Далее пишется постановка задачи, на основе выданного задания. Это должен быть связный текст в объеме до 1-1,5 страниц. В этом разделе необходимо раскрыть цели и задачи УИРа/диплома.

## 2. Разработка моделей и алгоритмов ...

В этой главе описываются разработанные/модифицированные модели/методы/ алгоритмы, или/и описывается применение известных стандартных методов. Также, в конце главы обычно приводится общая архитектура программной системы, вытекающая из описанной теории. Приведенные ниже заголовки подразделов так же весьма примерные и сильно зависят от особенностей конкретной работы.

Формулы и их части необходимо набирать в математическом режиме (символ  $\$$ ). Во избежание переноса длинных формул между строками их стоит размещать по центру колонки, например,

$$Sabc = (\lambda xyz.xz(yz))abc = ac(bc),$$

и, если абзац после формулы продолжается, необходимо использовать `\noindent`.

Для набора правил вывода можно использовать пакет `mathpartir.sty`. Правила вывода могут быть вынесены в виде рисунка (см. рис. 2.1).

### 2.1 Модель системы ...

...

### 2.2 Метод решения задачи для ...

...

### 2.3 Алгоритмы вычисления ...

...

### 2.4 Обобщенная архитектура и интерфейсы ...

В ряде случаев, все или некоторые результаты проектирования могут быть представлены во второй главе. Обычно же архитектура описывается в третьей главе.

### 2.5 Выводы

Необходимо перечислить, какие теоретические результаты были получены с указанием степени новизны. Например: «Была разработана такая-то модель. Она представляет собой адаптированную версию модели X, в которой уравнение Z заменено на уравнение Z'». Еще пример: «Была предложена такая-то архитектура, она отличается от типовой в том-то и том-то. Это позволяет избежать таких-то проблем.». При этом не следует заниматься «высасыванием из пальца»: «Поставленная задача является типовой; для ее решения применены стандартные средства (перечислить,

$$\begin{array}{ccc}
\frac{M \rightarrow M'}{NM \rightarrow NM'} & (\mu) & \frac{M \rightarrow M'}{MN \rightarrow M'N} \quad (\nu) \qquad \frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'} \quad (\xi)
\end{array}$$

Рис. 2.1 – Правила редукции

какие).».

### **3. Результаты проектирования ...**

В этой главе описывается, что и как было спроектировано. При необходимости, описывается использованная методика проектирования. Сюда же относится описание внешних и внутренних программных интерфейсов, а также форматы и структуры входных и выходных данных.

#### **3.1 Использование методики «такой-то» для проектирования программных систем «такого-то типа»**

...

#### **3.2 Общая архитектура системы ...**

...

#### **3.3 Архитектура подсистемы 1...**

...

#### **3.4 Архитектура подсистемы $N$ ...**

...

#### **3.5 Проектирование протокола взаимодействия подсистем $X$ и $Y$**

...

#### **3.6 Выводы**

Следует перечислить, какие инженерные результаты были получены, а именно: какие программные системы, подсистемы или модули были спроектированы. Следует не только назвать полученные архитектуры, но и отметить их отличительные особенности.

## 4. Реализация и экспериментальная проверка ...

В этой главе описывается, что и как было запрограммировано, отлажено, протестировано, и что в результате получилось. Большинство работ должны содержать приведенные ниже разделы. Но нужно учитывать, что точный состав этой главы, как и других глав, зависит от специфики работы.

Фрагменты программного кода в тексте необходимо выделять при помощи команды `\verb`. Многострочные листинги должны оформляться при помощи пакета `listingsutf8`. Пример:

```
# let s x y z = x z (y z);;
val s : ('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c = <fun>
# let k x y = x;;
val k : 'a -> 'b -> 'a = <fun>
# let i = s k k;;
val i : '_a -> '_a = <fun>
```

Листинг 4.1 иллюстрирует использование выносных листингов. Листинг 4.2 показывает пример включения внешнего файла в качестве листинга, в данном случае — выносного.

### 4.1 Выбор инструментальных средств

В этом разделе обосновывается выбор инструментальных средств; одним из критериев выбора могут быть какие-либо требования к разрабатываемой системе, и если этих требований много, они могут быть выделены в отдельный раздел, или же в приложение. Этот пункт не пишется, если в аналитической главе был раздел, посвященный сравнительному анализу и выбору инструментальных средств.

### 4.2 Состав и структура реализованного программного обеспечения

Нужно охарактеризовать реализованное ПО: является ли оно настольной программой для Windows, или веб-приложением в форме сайта/веб-сервиса, или модулем/подключаемой библиотекой, или .... Также нужно перечислить, из чего оно состоит: какие исполняемые файлы и их назначение, конфигурационные файлы, файлы баз данных, требования к программному и аппаратному окружению, и т.п.

Если реализованное приложение достаточно обширно, этот раздел может быть разделен на

---

```
List myList = new List();
Element myElement = new Element();
myList.Append(myElement);
```

---

Листинг 4.1 – Выносной листинг

---

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```

---

Листинг 4.2 – Листинг из файла HelloWorld.scala

несколько: один с общим описанием, и по одному на подсистемы самого верхнего уровня.

### 4.3 Основные сценарии работы пользователя

Нужно помнить, что пользователем может быть не только «менеджер» или «человек в белом халате», но и другой программист. Последнее относится, в первую очередь, к реализованным библиотекам. Для «обычных» приложений нередко бывают пользователи нескольких категорий — например, обычный пользователь и администратор. Для каждой категории нужно описать, как выполняются основные функции, предпочтительно, с помощью серии скрин-шотов. Однако считается плохим тоном вставлять длинную вереницу из скрин-шотов: если их много, большую часть нужно выносить в приложение. Для *этого* раздела нормальной является плотность скрин-шотов из расчета: 1 страница скрин-шотов на 1-2 страницы текста.

### 4.4 Разработка тестовых примеров

Описываются наиболее характерные тестовые примеры, для прогона на интеграционных тестах. (Да, использование unit-тестирования — это почти всегда хорошо, основное исключение составляют работы, в которых используемый инструментарий по какой-либо причине в принципе исключает такую возможность. Например, что-нибудь вроде Mathematica.)

В этом же разделе могут приводиться и результаты тестирования, включая таблицы и графики. Результаты тестирования могут быть вынесены в отдельный раздел, если много текстового материала и/или использована (не совсем) стандартная методика тестирования (описание которой также нужно привести).

**Замечание.** В ПЗ (как УИРа, так и ВКР) следует избегать ситуаций, когда значительную часть основного содержания составляют страницы с иллюстрациями и таблицами, особенно, если такие страницы следуют подряд. В основном тексте следует оставлять лишь самые основные таблицы и рисунки, а остальное — выносить в приложение.

## **4.5 Сравнение реализованного программного обеспечения с существующими аналогами**

В сравнении должно быть отражено, чем полученное ПО выгодно (и невыгодно) отличается от прочих ближайших аналогов. Практика показывает, что аналоги есть всегда. А если нет аналогов, значит есть частичные решения, которые реализуют какие-то части функционала вашей системы. Тут тоже может быть относительно много таблиц и графиков.

## **4.6 Выводы**

Следует перечислить, какие практические результаты были получены, а именно: какое программное или иное обеспечение было создано. В число результатов могут входить, например, методики тестирования, тестовые примеры (для проверки корректности/оценки характеристик тех или иных алгоритмов) и др. По каждому результату следует сделать вывод, насколько он отличается от известных промышленных аналогов и исследовательских прототипов.



## **Заключение**

В заключении в тезисной форме необходимо отразить результаты работы:

- аналитические (что изучено/проанализировано);
- теоретические;
- инженерные (что спроектировано);
- практические (что реализовано/внедрено).

Примерная формула такая: по каждому указанному пункту приводится по 3-5 результатов, каждый результат излагается в объеме до 5 фраз или предложений.

Также есть смысл привести предполагаемые направления для будущей работы.

Общий объем заключения не должен превышать 1,5 страниц (1 страницы для УИРов).

## Список литературы

1. ГОСТ Р 7.0.53-2007 Система стандартов по информации, библиотечному и издательскому делу. Издания. Международный стандартный книжный номер. Использование и издательское оформление. — М.: Стандартинформ, 2007. — 5 с.

## А. Основные правила форматирования

Текст пояснительной записки должен готовиться для печати на листах формата А4, использоваться должен шрифт с засечками (Roman; обычно — Times Roman или Times New Roman), 12 или 14 кегль. Размеры полей:

- верхнее: 20 мм.
- нижнее: 20 мм.
- левое: 10 мм.
- правое: 25 мм.

Нумероваться должны все страницы, начиная с первой (титульной), однако сами номера следует проставлять на страницах, начиная со страницы реферата. Номер следует проставлять внизу страницу (в центре).

Заголовки оформляются тем же шрифтом, что и основной текст (т.е., соответственно, Times Roman или Times New Roman). Для заголовков первого уровня размер шрифта может быть больше размера шрифта основного текста (обычно 14-16).

Все разделы текста: реферат, оглавление, введение, три главы основного содержания, список литературы, заключение, приложения — должны снабжаться содержательным заголовком и начинаться с новой страницы; сами заголовки следует при этом центрировать (заголовки параграфов и пунктов выравниваются по ширине). Следует обратить внимание, что заголовки всех разделов, кроме трех основных глав, регламентированы; заголовки трех основных глав должны быть содержательными и отражать суть соответствующей главы. Названия типа «Аналитическая часть» и «Теоретическая глава» — *недопустимы*.

Текст пояснительной записки может содержать рисунки и таблицы. Все рисунки и таблицы должны снабжаться номерами и подписями:

- нумерация рисунков и таблиц должна быть сквозная (но отдельная, т.к. для рисунков своя, для таблиц — своя);
- в случае большого количества иллюстраций/таблиц, допускается «вложенная» нумерация (т.е. таблицу/рисунок можно снабжать составным номером в формате

⟨номер главы⟩.⟨номер внутри главы⟩;

- подрисуночная подпись должна располагаться снизу по центру;
- название таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире (ГОСТ 7.32-2001, п.6.6.1).

Здесь перечислены не все, а лишь основные требования к оформлению. Прочие требования — см. соответствующие ГОСТы.

Если рецензент ругается на большие отступы в списках, следует использовать `compactitem` и `compactenum` из пакета `paralist`. Например:

- это;
- не нумерованный;
- список;
- без лишних промежутков.

И для нумерованных списков:

- 1) нумерованные списки;
- 1) пакета `paralist`;
- 1) еще и удобно настраивать;
- 1) (например, менять формат номера).

## **В. Общая структура пояснительной записки**

1. Титульный лист
2. Лист с подписями (только для ВКР)
3. Задание (в данном примере используется задание на диплом)
4. Реферат (всегда на отдельной стр.)
5. Оглавление. Начинается с новой страницы.
6. Введение
  - 6.1. Актуальность
  - 6.2. Новизна
  - 6.3. Оригинальная суть исследования
  - 6.4. Содержание ПЗ по главам (тезисно)
7. Аналитическая глава. Пишется в стиле *аналитического обзора*
8. Теоретическая и инженерная глава. Описываются использованные, доработанные и разработанные модели, алгоритмы, методы, и т.п. Кроме того, тут формулируется архитектура системы.
9. Инженерная и практическая глава. Описывается реализация, а также остальные инженерные вопросы, не описанные в гл. 2. Примерное содержание такое
  - 9.1. Состав и структура реализованного ПО
  - 9.2. Выбор инструментальных средств
  - 9.3. Основные сценарии работы различных категорий пользователей
  - 9.4. Результаты тестирования (разработка тестовых примеров, таблицы и графики результатов прогона)
  - 9.5. Сравнение с существующими аналогами
10. Заключение

## 11. Список литературы

## 12. Приложения

Кроме того, в ПЗ могут включаться и такие разделы, как словарь терминов, список сокращений и др. В зависимости от предпочтений автора, могут помещаться как в начале ПЗ (до оглавления), так и в конце (после заключения, но до приложений).

### **Замечания:**

1. На каждый элемент из списка литературы должна быть хотя бы одна ссылка в тексте.
2. Список литературы должен быть оформлен согласно ГОСТ [1].
3. Минимальное количество источников для УИРов — 15–20 (для работ с большой аналитической и теоретической частью нормальное количество — 25–30 и более), для дипломов — соответственно, 30–35 и 35–60. Эти цифры существенны, т.к. «недобор», как правило, свидетельствует о не выполнении аналитической части работы и, следовательно, недостаточном владении предметом.
4. При подготовке РСПЗ рекомендуется вставлять уже наработанные к моменту подготовки РСПЗ материалы. Однако, в любом случае, каждый раздел должен начинаться с аннотации, заключенной в окружение \annotation. В пояснительной записке к диплому аннотации не нужны.
5. Между заголовком главы и первым разделом рекомендуется поместить один-два абзаца связанного текста с кратким содержанием (планом) главы.
6. Общее число и объем приложений не ограничивается. Объем ПЗ *без* приложений — 25–40 стр. для УИРов, и не менее 60–100 стр. для дипломов. Объем ПЗ не может быть меньше указанных размеров. Это означает, что студент не выполнил работу, или, как минимум, не удосужился подготовить адекватную ПЗ. Превышать верхние пределы также не желательно, в некоторых комиссиях это может восприниматься негативно; однако, в целом, небольшое превышение допустимо, если проделана действительно большая работа и получено много результатов (например, экспериментальных, или получены нетривиальные аналитические или теоретические результаты).
7. ГОСТ требует, чтобы нумерация страниц начиналась с первой, титульной, страницы. При на самой титульной странице номер не печатается. В данном случае, номера также не про- ставляются на листах задания, а также на листе с подписями (для ВКР).

## С. Правила использования шаблона

Настоящий шаблон все еще несколько несовершенен в плане оформления: например, неправильная нумерация приложений, и еще несколько нюансов. В последующих версиях это будет исправляться.

Ниже описана структура исходных текстов шаблона (и, соответственно, структура исходных текстов ПЗ).

Головной файл — `thesis-template.tex`. Его задача — «склеить» вместе разные части ПЗ. Каждая часть (реферат, введение, каждая содержательная глава, заключение, библиография, приложения) выделяется в отдельный файл.

`thesis-macro.tex` — содержит определения различных макрокоманд, которые часто используются в конкретной работе, например, определения окружения для теорем, некоторые часто используемые формулы, и т.п.;

`thesis-abstract.tex` — содержит аннотацию;

`thesis-intro.tex` — содержит введение;

`thesis-chapter1.tex` — текст первой главы;

`thesis-chapter2.tex` — текст второй главы;

`thesis-chapter3.tex` — текст третьей главы;

`thesis-bib1.tex` — список литературы (только подключение к проекту);

`biblio.bib` — собственно библиография (в формате BibTeX);

`thesis-conclusion.tex` — заключение;

`thesis-appendix1.tex` — первое приложение;

`thesis-appendix2.tex` — второе приложение;

Головной файл нужно менять лишь тогда, когда нужно добавить в проект новый файл, или удалить существующий (см. команду `\input`). Обычно, это требуется, когда нужно добавить/удалить приложения.

Для того, чтобы L<sup>A</sup>T<sub>E</sub>X при компиляции автоматически «подхватил» задание, его нужно сохранить в формате pdf (например, с помощью виртуального принтера), поместить в ту же папку /title и назвать task.pdf. Точно также следует поступить с титульной страницей (title.pdf). При оформлении ПЗ для ВКР следует дополнительно поместить в папку /title pdf-версию листа с подписями, назвав файл title-dep22.pdf. После этого, нужно раскомментировать команду \includepdf[pages={-}, offset=0mm -0mm]{title/title-dep22.pdf} в начале головного файла. Образцы и Word-шаблоны титульных листов для (РС)ПЗ к УИРам, НИРам, практикам и ВКР лежат в папках /title/магистры и /title/бакалавры.

В этой версии шаблона используется BibTeX, поэтому для оформления списка литературы используются два файла: thesis-bibl.tex и biblio.bib. Использование BibTeX дает ряд преимуществ. Не нужно заботиться о порядке сортировки, это делается автоматически; не нужно заботиться, на какие элементы библиографии есть ссылки — печатаются только использованные в тексте элементы. Кроме того, многие курсовые проекты выполняются на протяжении ряда лет. С BibTeX проще собирать список литературы и управлять им.

**Замечание.** В шаблоне используется пакет hyperref, который делает две вещи: все перекрестные ссылки «кликабельны», а также выделены (красной) рамочкой. Эти рамки *не выводятся на печать*. Вместо цветных рамок, возможны другие способы выделения ссылок (см. документацию пакета).