

Направление подготовки  
**09.03.04 Программная инженерия**

Направленность (профиль) образовательной программы  
**Разработка программно-информационных систем**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ  
«КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

ВВЕДЕНИЕ .....	2
1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ .....	2
2. СОДЕРЖАНИЕ И ОБЪЕМ КУРСОВОЙ РАБОТЫ .....	2
2.1 Требования к реализуемому программному продукту (программе) .....	3
2.2 Защита курсовой работы .....	5
2.3 Структура данных на сдаваемом носителе.....	6
3 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ.....	7
4 ОСНОВНОЕ СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ.....	13
5. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ НЕКОТОРЫХ РАЗДЕЛОВ И ГЛАВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ .....	14
5.1 Пояснение к содержанию первой главы .....	14
5.2 Пояснение к содержанию пункта посвященного описанию структуры программного продукта .....	15
5.2 Пояснение к содержанию пункта алгоритмической реализации.....	15
5.3 Пояснение к содержанию пунктов содержащих руководства программиста и пользователя .....	15
5.3 Пояснение к содержанию пункта тестирования разработанного программного продукта .....	17
6. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ПРИЛОЖЕНИЯ «ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА» .....	17
7. ПРИМЕРНЫЕ ТЕМАТИКИ ДЛЯ КУРСОВОЙ РАБОТЫ .....	18
Приложение 1 .....	21
Приложение 2 .....	22
Приложение 3 .....	23
Приложение 4 .....	24

## ВВЕДЕНИЕ

В курсовой работе по дисциплине «Конструирование программного обеспечения» рассматриваются вопросы алгоритмической постановки и программной реализации системы обработки данных с внешними подключаемыми модулями. В ходе выполнения курсовой работы студенты приобретают углубленные знания о методах конструирования программного обеспечения, о способах организации взаимодействия приложений, а также приобретают навыки разработки динамических библиотек и самостоятельной работы.

### 1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Основными целями курсовой работы по дисциплине «Конструирование программного обеспечения» являются систематизация, закрепление и расширение теоретических знаний по специальности, и применение их для решения задач обработки данных, а также получение навыков разработки приложений использующих динамические библиотеки для расширения базового функционала приложения. Также выявление степени подготовленности студента к самостоятельной работе, его творческого потенциала.

### 2. СОДЕРЖАНИЕ И ОБЪЕМ КУРСОВОЙ РАБОТЫ

Курсовая работа выполняется студентами в течение всего времени обучения по дисциплине «Конструирование программного обеспечения» в объеме 36 часов. Выполнение курсовой работы заключается в разработке программного обеспечения ориентированного на обработку данных с применением динамически подключаемых библиотек расширяющих функциональные возможности и оформлении пояснительной записки к курсовой работе в соответствии с методическими указаниями по выполнению данной курсовой работы.

Объем пояснительной записки составляет 30-45 страниц основного текста на листах формата А4 (без учета приложений). Примерное содержание пояснительной записки (*может измениться в зависимости от выбранной темы*):

Титульный лист

Задание на курсовую работу

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор (Анализ) решаемой задачи/аспектов предметной области

1.2 Обзор основных систем (**по предметной области**)

1.3 Обзор технологий **необходимых для решения задачи и реализации ПП**

1.4 Обзор методов и алгоритмов (**ключевые алгоритмы согласно предметной области, а также алгоритмы необходимые для защиты разрабатываемого ПП**)

1.5 Выводы по главе

## 2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

### 2.1 Структура программного продукта

#### 2.1.1 Система/модуль А

#### 2.1.2 Система/модуль \*\*\*\*

#### 2.1.N Организация системы плагинов

### 2.2 Алгоритмическая реализация

#### 2.2.1 Функциональный блок Б1

#### 2.2.2 Функциональный блок \*\*\*\*

### 2.3 Руководство программиста

### 2.4 Краткое руководство пользователя

### 2.5 Тестирование

### 2.6 Выводы по главе

## ЗАКЛЮЧЕНИЕ

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

## ПРИЛОЖЕНИЯ

Приложение А. Техническое задание на разработку программного продукта

Приложение Б. Примеры \*\*\*\*

Целесообразно выбор темы курсовой работы согласовывать с темой предполагаемой выпускной квалификационной работы (ВКР).

При реализации программы в рамках курсовой работы студент должен по возможности (в зависимости от тематики) предложить и использовать адаптивную обработку данных в зависимости от подключенных модулей расширения (плагинов).

Необходимо предоставить детальное описание реализуемых алгоритмов и схем взаимодействия между модулями с использованием графических схем и/или *UML*-диаграмм.

### **2.1 Требования к реализуемому программному продукту (программе)**

В программе должна быть реализована система модулей расширения (плагинов) позволяющая выполнять следующие действия:

- автоматическое обнаружение модулей расширения в определенной директории и добавление их функционала в основную программу при запуске;
- отображение списка подключенных модулей расширения с выводом информации о них [название; тип; авторство; версия; дата релиза];
- контроль загрузки плагинов на основе уникального идентификатора (GUID), для исключения возможности повторной загрузки одного и того же плагина;
- настройка параметров работы алгоритма через динамически формируемый приложением графический интерфейс.

Для решения основной задачи предусмотреть возможность задания цепочки использования плагинов и/или алгоритмов обработки данных. Так элементы цепочки могут представлять собой функциональные действия по получению данных с различных источников, их обработку и сохранение результата.

Требования к реализуемому программному продукту в зависимости от оценки:

На оценку «отлично»	
Параметр	Значение
Всего плагинов не менее * из которых:	<b>6</b>
** с графическим интерфейсом настройки параметров алгоритма	<b>3</b>
Настройка цепочки обработки, с задействованием не менее * разных алгоритмов	<b>3</b>
Разграничение пользователей (групповая политика) для * категорий <b>требование может не учитываться при отсутствии в разрабатываемом приложении/системе групповых политик в явной форме, в таком случае будет полнофункциональная и демонстрационная версии</b>	<b>3</b>
Подключение/отключение плагинов во время выполнения программы	
Возможность сохранения/загрузки очереди задач на обработку НД с учётом настройки параметров работы алгоритмов (сохранение в файл произвольного формата). <sup>1</sup>	
Возможность сохранения/загрузки конфигурации подключенных модулей расширения (плагинов). Конфигурационные данные сохраняются в файле, например *.ini файле;	
Система защиты приложения на основе USB Flash, а также данных в реестре Windows или особом файле в каталоге с программой	
Программный модуль с генерацией ключей (с использованием механизмов кодирования данных с задействованием криптографических алгоритмов) для защиты на основе USB Flash позволяющий вводить следующие виды ограничений: <ul style="list-style-type: none"> <li>– время работы (указывается количество дней или конечная дата)</li> <li>– функциональности (блокировка части интерфейса, и ограничение по времени работы)</li> <li>– профиль пользователя (групповая политика) *</li> </ul> Файл ключа должен содержать сведения о пользователе которому выдана лицензия. При этом данные ключа должны шифроваться с учетом сведений USB Flash и осуществляться контроль целостности ключа должен быть на основе хэш-кода.	
Модуль протоколирования встроенный в основное программное приложение на основе базы данных, и отдельное приложение (независимы исполняемый файл) с возможностью визуализации данных в форме графиков/отчетов!! <sup>2</sup>	
На оценку «Хорошо»	
Всего плагинов не менее * из которых:	<b>5</b>
** с графическим интерфейсом настройки параметров алгоритма	<b>3</b>
Настройка цепочки обработки, с задействованием не менее * разных алгоритмов	<b>3</b>
Разграничение пользователей (групповая политика) для * категорий <b>требование может не учитываться при отсутствии в разрабатываемом приложении/системе групповых политик в явной форме, в таком случае будет полнофункциональная и демонстрационная версии</b>	<b>2</b>
Подключение/отключение плагинов во время выполнения программы	
Система защиты приложения на основе USB Flash	
Программный модуль с генерацией ключей для защиты на основе USB Flash (с кодированием на основе BASE64 или на базе криптографических алгоритмов, с хеширование данных накопителя, например при помощи MD5) позволяющий вводить следующие виды ограничений: <ul style="list-style-type: none"> <li>– время работы (указывается количество дней или конечная дата)</li> <li>– профиль пользователя (групповая политика)*</li> </ul> В файле ключа также должны указываться параметры пользователя	
Модуль протоколирования встроенный в основное программное приложение, с возможностью визуализации данных в форме графиков/отчетов!! <sup>2</sup>	

<sup>1</sup> Исходя из тематики курсовой работы данный пункт может быть необязательным

<sup>2</sup> В качестве протоколируемых данных могут быть сведения о работе программы, данные по запускам, по использованию функционала, данные о времени работы алгоритмов и т.п.

На оценку «Удовлетворительно»	
Параметр	Значение
Всего плагинов не менее * из которых: ** с графическим интерфейсом настройки параметров алгоритма	4
	2
Настройка цепочки обработки, в которой задействованы не менее * разных алгоритмов	2
Наличие режима разграничения функционирования – полноценная и демонстрационная версия	
Система защиты приложения, на основе файла ключа реализуемая в виде модуля интегрированного в исполняемый файл.	
Программный модуль с генерацией ключей определяющих параметры работы программы.	
Модуль протоколирования действий встроенный в основное программное приложение, с возможностью визуализации данных в форме графиков/отчетов! <sup>3</sup>	

**В зависимости от выбранной предметной области требования для получения оценок могут меняться. В связи с этим они обговариваются индивидуально.**

При формировании оценки также влияет способ организации работы с базой данных и взаимодействия с дополнительными модулями программ. Также повышающими оценку факторами являются:

- реализация возможности получения наборов данных через интернет (HTTP|FTP) и/или буфер обмена;
- пакетная обработка данных с использованием возможности параллельной обработки данных (выполнение цепочки алгоритмов обработки для разных НД в отдельных высокоуровневых потоках TThread или процессах);
- реализация механизмов получения исходных данных из сторонних приложений (т.е. приложений разработанных не обучающимся);
- реализация механизмов защиты программного продукта на основе USB Flash с учётом введения разноплановой проверки (Trial режим работы и ограничения в ключах) с применением кодирования данных приложения (например: конфигурация плагинов, настройки работы программы, параметры доступа к сервисам и т.п.);
- организация экспорта данных по запросу в файлы MS Excel, MS Word и/или PDF;

## **2.2 Защита курсовой работы**

До защиты студент должен получить допуск к защите курсовой работе по результатам проверки преподавателем пояснительной записки и демонстрации программного продукта путем компиляции исходных текстов. Защита включает доклад (3 – 5 минут) о проделанной работе с демонстрацией программного продукта по ходу доклада и ответы на вопросы.

Для проставления оценки студент должен сдать преподавателю пояснительную записку к курсовой работе и диск (CD или DVD) или USB накопитель с программой и сопутствующими материалами. Помимо этого необходимо выложить в портфолио (<https://portfolio.pallada.sibsau.ru/>) пояснительную записку в формате **pdf**, где первые две страницы будут представлены сканами с подписями.

<sup>3</sup> В качестве протоколируемых данных могут быть сведения о работе программы, данные по запускам, по использованию функционала, данные о времени работы алгоритмов и т.п.

### 2.3 Структура данных на сдаваемом носителе

На сдаваемом носителе должна быть записана следующая информация:

#### ДИСК/ USB накопитель:

- Текстовый файл «**информация.txt**» Файл должен содержать сведения о разработчике (группа, ФИО), а также информацию по курсовой работе (дисциплина, наименование КР), При этом в случае использования групповых политик или наличия особых параметров для эксплуатации в файле должны быть отражены соответствующие данные.
- Программа (Программа со всеми необходимыми файлами для её запуска. **!!! После копирования программы на жесткий диск компьютера, она должна нормально функционировать**)
  - База данных/СУБД (**Опционально** – при наличии портативной версии СУБД, например MySQL)
  - Plugins (Каталог содержащий плагины для разработанного ПП)
  - Экспериментальные данные (Данные использованные для проведения экспериментов)
  - KeyGen (Каталог содержащий генератор ключей безопасности/защиты ПП)
- Инсталлятор (Инсталляционный комплект, позволяющий полноценно развернуть программный продукт на компьютере без дополнительных действий)
- Исходные коды (Исходные коды программы)
  - Исходные коды основной программы (Исходные коды программы)
  - Исходные коды дополнительных модулей (Исходные коды модулей программы с выделением для каждого модуля отдельного каталога)
  - Исходные коды плагинов (Исходные коды плагинов с выделением для каждого плагина отдельного каталога)
  - Исходные коды KeyGen (Исходные коды генератора файла ключа)
  - Исходные коды модуля защиты (Исходные коды модуля или windows сервиса для организации механизма защиты)
  - Библиотеки и компоненты (компоненты и библиотеки необходимые для компиляции программы, не включенные в состав среды разработки)
- Пояснительная записка (Пояснительная записка в форматах \*.docx, \*.pdf)
- Дополнительные материалы (Материалы необходимые для нормального функционирования программы устанавливаемые отдельно)
  - Драйверы
  - RunTime Библиотеки
  - Другое

### Отчетность студентов на промежуточную аттестацию

1-ая аттестация	Выбор темы (в течении 2х недель)
	Краткое техническое задание (в течении 2х недель)
	Описание интерфейса организации системы плагинов (включая виды плагинов)
	Диаграммы активности проверки и подключения плагина в систему
	Примерная схема описания потоков данных при взаимодействии с плагинами
	Примерная структура функциональных модулей разрабатываемого программного продукта
	Диаграммы активности (алгоритмов) в зависимости от сути темы не менее 2х
	Написание частей пояснительной записки
	Элементы первой главы (заготовки для пунктов 1.1 и 1.2)

2-ая аттестация	Техническое задание с полным описанием функциональных требований
	Плагины, реализующие не менее двух вычислительных алгоритмов (по предметной области)
	Программное обеспечение, с частичной функциональностью: – подключение модулей расширения (плагинов); – отображение списка подключенных плагинов с выводом информации о них (название; авторство; версия; дата релиза); – настройку одного алгоритма содержащегося в плагинах через графический интерфейс; – выполнение алгоритма обработки данных из плагина
	Написание частей пояснительной записки (Элементы первой главы (п. 1.1-1.5) и частично второй главы (п. 2.1 и п. 2.2) .

### 3 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

Материал, включаемый в курсовую работу, должен быть обработан и систематизирован. Общими требованиями к тексту курсовой работы являются:

- четкость и логическая последовательность изложения материала;
- убедительность аргументации;
- краткость и точность формулировок, исключающих возможность неоднозначного толкования;
- конкретность изложения результатов работы;

**Текст пояснительной записки** должен быть тщательно отредактирован, набран на компьютере в текстовом редакторе. Размещается текст на одной стороне листа формата А4 (210×297), при этом соблюдая следующие размеры полей: левое – 25 мм; правое – 15 мм; верхнее – 20 мм; нижнее – 20 мм.

Текст следует печатать шрифтом **Liberation Serif**, параметры стиля основного текста:

*Размер шрифта* – 14 пунктов.

*Расстановка переноса* – автоматическая, ширина зоны переноса слов – 0,25.

*Выравнивание текста* – по ширине.

*Абзацный отступ* – 1,25 см.

*Интервалы до и после абзаца* – 0 пт.

*Межстрочный интервал* – одинарный.

В зависимости от текста меняется начертание. Так русские обозначения и греческие символы имеют **прямое** начертание, а латинские символы – **курсив**.

Англоязычные слова и отдельные буквы на английском языке выделяются курсивом как по всему тексту, так и в подрисуночных подписях и названиях таблиц.

В тексте курсовой работы, за исключением формул, таблиц, рисунков, не допускается:

- применять математический знак «минус» (–) перед отрицательными величинами (следует писать слово «минус»);
- применять без числовых значений математические знаки, например > (больше), < (меньше), = (равно), ≥ (больше или равно), ≤ (меньше или равно), ≠ (не равно), а также знаки № (номер), % (процент).

**Нумерация страниц** в работе должна быть сквозной. На титульном листе и бланке задания на курсовую работу номер страницы не ставится, но

включается в общую нумерацию. Первая страница разделов верхнего уровня печатается без отображения нумерации (номер страницы можно скрыть белым прямоугольником помещенным поверх текста). Нумерация страниц проставляется в нижнем колонтитуле по центру. Номер страницы проставляется без точки, от края до колонтитула – 14 мм, размер колонцифры – 12 п. ко всему тексту.

**Титульный лист** является первой страницей курсовой работы. Оформление титульного листа **см. приложение 1**.

**Задание на курсовую работу** является второй страницей работы. Бланк задания заполняется студентом в рукописной форме. **В приложение 2** приведен бланк задания на курсовую работу.

### Оформление заголовков

Текст курсовой работы делят на главы (разделы), параграфы (подразделы) и пункты. Каждую главу (раздел) следует начинать с нового листа (страницы).

Заголовки разделов, подразделов и пунктов следует печатать с абзацного отступа без точки в конце, при этом переносы слов в заголовках не допускаются. Параметры стилей для заголовков:

Свойство стиля	Заголовки			
	1-ого уровня	2-ого уровня	3-го уровня	4+ уровня
Выравнивание	По левому краю			
Отступ слева	1,25 см.			
Межстрочный интервал	1 (Одинарный)			
Интервал до абзаца	0 пт.	16 пт. / 0 пт.*	0 пт.	
Интервал после абзаца	32 пт.	16 пт.	0 пт.	
Дополнительно	Не добавлять интервал между абзацами одного стиля			
Регистр	Прописные буквы	Как в предложениях		
Начертание	Полужирный	Полужирный		Курсив
Дополнительные параметры: – Разбивка на страницы – Исключение форматирования	С новой страницы			
	Запрет висячих строк Не отрывать от следующего Не разрывать абзац Запретить автоматический перенос слов			

\* – значение 0 пт. необходимо задавать в том случае, если заголовок второго уровня идет непосредственно за заголовком первого уровня и текстовый редактор не поддерживает автоматическую подстройку параметров интервала до/после абзаца.

Нумеровать разделы, подразделы и пункты следует арабскими цифрами. Разделы должны иметь порядковую нумерацию в пределах всего текста, за исключением содержания, введения, заключения, списка использованных источников и приложений, которые не нумеруются. При оформлении заголовков нумеровать и выносить в содержание следует только заголовки до третьего уровня включительно.

После номера раздела, подраздела и пункта точка не ставится. В случае если раздел или подраздел имеет только один пункт, то нумеровать и выносить в содержание его не нужно.

**Рисунки.** Рисунки размещаются непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все рисунки должны быть даны ссылки.

Каждому рисунку или таблице присваивается номер по порядку упоминания в тексте каждого раздела отдельно. Первая цифра означает принадлежность рисунка к разделу, а вторая – к записи. В тексте необходимо делать ссылки на все рисунки и таблицы. Параметры стиля для блока рисунка:

*Шрифт: Liberation Serif*

*Размер шрифта – 12 пунктов.*

*Выравнивание – по центру.*

*Отступ слева – 0 см.*

*Интервал до абзаца – 16 пт.*

*Интервал после абзаца – 2 пт.*

*Межстрочный интервал – одинарный.*

*Дополнительно – не добавлять интервал между абзацами одного стиля.*

*Разбивка на страницы – запрет висячих строк, не отрывать от следующего.*

*Исключение форматирования – запретить автоматический перенос слов.*

Под каждым рисунком должна быть подпись и/или пояснительные данные (подрисуночный текст), отделенные от номера рисунка тире. Параметры стиля для подписи рисунка:

*Шрифт: Liberation Serif*

*Размер шрифта – 12 пунктов.*

*Выравнивание – по центру.*

*Отступ слева – 0 см.*

*Интервал до абзаца – 0 пт.*

*Интервал после абзаца – 16 пт.*

*Межстрочный интервал – одинарный.*

*Дополнительно – не добавлять интервал между абзацами одного стиля.*

*Разбивка на страницы – запрет висячих строк.*

*Исключение форматирования – запретить автоматический перенос слов.*

В случае, когда текст подрисуночной подписи не уместается в одну строку, то его следует разбить исходя из смыслового контекста на несколько строк, используя разрыв строки.

Если рисунок располагается на нескольких страницах (например, диаграмма активности), то на следующих страницах пишется номер рисунка и слово «Продолжение». Например, «Рисунок 2.3 – Диаграмма активности алгоритма», на последующих страницах – «Рисунок 2.3 – Продолжение». В случае если схема/диаграмма активности расположены на нескольких страницах, то на линиях разрыва необходимо проставить нумерацию (в окружности). Экранные формы оформляются как рисунки.

При ссылках на рисунки допускается делать ссылки внутри предложения «...представлено на рисунке 2.1». В случаях если для ссылки на рисунки нет явного поясняющего текста, то следует указывать ссылку в круглых скобках. В этом случае ссылка начинается с прописной буквы, например: «Диаграмма активности алгоритма (Рисунок 2.5) характеризует...».

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначение приложения. Например: Рисунок Б.3.

Для формирования UML диаграмм и ER-модели возможно использование программы **Software Ideas Modeler** (доступна на сайте <https://www.softwareideas.net/>). В случае использования данной программы при экспорте изображения необходимо установить белый цвет фона.

**Таблицы** в пояснительной записке должны быть компактными, иметь нумерацию по принципу рисунков, наименование, названия граф, размерность указываемых величин. Параметры базового стиля шрифта в таблице:

*Шрифт: Liberation Serif*

*Размер шрифта данных в таблице – 12 пунктов.*

*Отступ слева – 0 см.*

*Интервал до абзаца – 0 пт.*

*Интервал после абзаца – 0 пт.*

*Межстрочный интервал – одинарный.*

*Выравнивание* – определяется тем, что содержится в таблице. Так возможны варианты: по центру/по центру по левому краю /сверху по левому краю.

*Исключение форматирования* – запретить автоматический перенос слов.

Название таблицы следует помещать над таблицей слева и оформлять 12 пунктов без абзацного отступа в одну строку с ее номером и названием через тире, без точки в конце. Например, «Таблица 1.2 – Сравнение программных средств». В тексте производится упоминание в виде (табл. 1.2) или таблица 1.2. При разрыве таблицы на несколько страниц в качестве названия таблицы продолжения пишется, например, «Продолжение таблицы 1.2» и размещается по правому краю. В случае если на новую страницу переносится конечная часть таблицы и таблица размещена более чем на двух страницах, то в названии пишется, например, «Окончание таблицы 1.2».

Параметры стиля названия таблицы:

*Шрифт: Liberation Serif*

*Размер шрифта подписи таблицы – 12 пунктов.*

*Выравнивание – по левому краю (для продолжения – по правому краю)*

*Отступ слева – 0 см.*

*Интервал до абзаца – 4 пт.*

*Интервал после абзаца – 4 пт.*

*Межстрочный интервал – одинарный.*

*Дополнительно* – не добавлять интервал между абзацами одного стиля.

*Разбивка на страницы* – запрет висячих строк, не отрывать от следующего, не разрывать абзац.

*Исключение форматирования* – запретить автоматический перенос слов.

Заголовки столбцов и строк таблицы следует писать с прописной буквы в единственном числе, подзаголовки столбцов – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставятся.

Параметры шрифта заголовка столбцов и строк в таблице принимают аналогичные значения базовому стилю и добавляются следующие параметры:

*Выравнивание для столбцов* – по центру.

*Выравнивание для строк* – по левому краю.

*Разбивка на страницы* – запрет висячих строк, не разрывать абзац.

Не допускается разрыв ячеек на страницах. В случае, если таблица расположена на нескольких страницах, то выполняется разрыв таблицы и добавляются идентификаторы заголовков столбцов таблицы на каждой странице в виде числовых обозначений или полного копирования шапки столбцов.

При осуществлении разрыва таблиц, для обозначения того что на следующей странице будет продолжение, следует закрывающую линию таблицы сделать белым цветом:

Таблица 1.1 – Пример первой части

Название 1	Название 2	Название Б

Продолжение таблицы 1.1

Название 1	Название 2	Название Б

После таблицы необходимо оставить пустую строку с одинарным межстрочным интервалом (*стиль «Строка после таблицы»*). Пункт или раздел не должны заканчиваться таблицей.

**Математические формулы.** Набор формул следует осуществлять с помощью редактора Microsoft Equation (меню Вставка→Объект) с базовым размером 12 пт., единообразно по всему тексту курсовой работы.

Математические формулы и уравнения, вынесенные в отдельные строки, отделяются от текста сверху и снизу пробелом в одну строку и, как правило, располагаются посередине строки.

Если формула не умещается в одну строку, то она должна быть перенесена после знаков равенства (=), плюса (+), минуса (-), умножения (×), деления (:) или других математических знаков, причем знак в начале следующей строки повторяется.

Пояснения символов и коэффициентов, входящих в формулу, если они не сделаны ранее в тексте, должны быть помещены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться со слова «где» без двоеточия после него и без абзацного отступа в начале строки.

Формулы, следующие одна за другой и не разделенные текстом, отделяют запятой. В конце формулы и в тексте перед ней знаки препинания ставятся в

соответствии с правилами пунктуации. Двоеточие ставится, если в тексте перед формулой содержится обобщающее слово.

При указании ссылки на формулы в основном тексте работы формулы следует нумеровать используя двухуровневую нумерацию арабскими цифрами, заключенными в круглые скобки, которая состоит из номера раздела и порядкового номера формулы, разделенных точкой, например: (2.3), и располагать в крайнее правое положение строки. Формулы, помещенные в приложениях, должны нумероваться отдельно арабскими цифрами в пределах каждого приложения с добавлением перед каждой цифрой обозначения приложения, например: «формула (В.1)».

**Листинги** используются в тексте курсовой работы для отображения программного кода на языке программирования. При этом соответствующий блок кода предваряется словом «Листинг» с указанием его порядкового номера с соблюдением общих принципов нумерации (оформляется по аналогии с названием таблицы). Для набора текста программного кода используется следующий стиль «Листинг (фрагмент кода)»:

*Шрифт: Courier New*

*Размер шрифта – 12 пунктов.*

*Выравнивание – по левому краю*

*Отступ слева – 0 см.*

*Интервал до абзаца – 4 пт.*

*Интервал после абзаца – 4 пт.*

*Межстрочный интервал – одинарный.*

*Дополнительно – не добавлять интервал между абзацами одного стиля.*

**В список использованных источников** допускается включать не только издания, которые были фактически использованы автором, но и названия работ, отвечающих тематике представляемой КР, по которым автор проводил обзор. При написании работы студент обязан давать ссылки на источник, из которого он заимствует материалы. Помимо печатных изданий на русском языке могут использоваться материалы на иностранных языках и электронные ресурсы, при указании которых необходимо соблюдать следующий порядок:

- печатные издания на русском языке;
- печатные издания на иностранных языках;
- электронные ресурсы на русском языке;
- электронные ресурсы на иностранных языках.

Внутри каждого из перечисленных блоков списка использованных источников элементы оформляются в алфавитном порядке по фамилии первого автора либо по названию (при отсутствии авторов).

**Ссылка на литературу** в тексте пояснительной записки дается в виде цифр в квадратных скобках, где указывается номер книги (журнала) или интернет источника в списке используемых источников и страница для справочного издания.

Если авторов не более трех, то указываются фамилии, имена, отчества авторов, полное название книги или статьи (для статьи наименование журнала и номер журнала), место издания, наименование издательства, год издания, количество страниц.

Если авторов более трех, указываются первые три автора и оборот «и др.». Если книга издана под редакцией авторов, то вначале указывается ее полное библиографическое название. В списке литературы такая книга располагается в алфавитном порядке по своему наименованию.

Пример оформления списка использованных источников:

- 1) Гонсалес, Р., Вудс, Р. Цифровая обработка изображений. Пер. с англ. под ред. П. А. Чочиа. – М. : Техносфера, 2012. – 1072 с.
- 2) Обработка изображений: Быстрые алгоритмы шумоподавления [Электронный ресурс]. – URL: [http://www.neкое\\_url.ru/article14/](http://www.neкое_url.ru/article14/) (дата обращения: 29.01.2025).

#### 4 ОСНОВНОЕ СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

**Во введении** определяется предмет исследования, дается общая характеристика курсовой работы, обоснование актуальности темы, состояние разработки проблемы, о которой пойдет речь в работе, основные цели и задачи, практическая значимость и описания состава курсовой работы. Объем введения около 2-3 страниц.

**Основная часть** состоит из двух глав. Первая глава, как правило, носит теоретическо-методологический характер. Здесь нужно: проанализировать предметную область: выполнить обзор основных систем, в которых имеется необходимый функционал для решения поставленной задачи, а также выявить их сильные и слабые стороны; рассмотреть технологии, средства для разработки (библиотеки/компоненты), а также возможные для использования при реализации методы и алгоритмы. Содержание второй главы носит практический характер, здесь описывается реализация разрабатываемого программного продукта и тестирование разработанного программного продукта и/или отдельных алгоритмов.

**Заключение** содержит **выводы**, которые должны быть общими по всей работе и отражать действия, выполненные студентом в ходе выполнения курсовой работы. Они должны содержать общие выводы, обобщенное изложение основных проблем, авторскую оценку работы с точки зрения решения задач, поставленных в курсовой работе.

После выводов дается **список использованных источников**. Список организуется в соответствии с требованиями библиографического описания произведений печати и электронных изданий. Библиографический список должен состоять **не менее чем из 15 источников**.

**Приложения** помещают после списка использованных источников. Помимо обязательного приложения «Техническое задание на разработку программного продукта» могут быть дополнительные, в которые помещается часть материала основных глав. Так в приложения могут быть размещены таблицы, схемы, графики, статистические материалы, графическая часть курсовой работы, в том числе распечатки фрагментов псевдокода или SQL кода запросов и т. п., но при этом не должно нарушаться восприятие глав.

## 5. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ НЕКОТОРЫХ РАЗДЕЛОВ И ГЛАВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

### 5.1 Пояснение к содержанию первой главы

В данной главе необходимо изложить информацию по выбранной предметной области, которая должна включать следующие пункты

**Обзор (Анализ) решаемой задачи/аспектов предметной области** – в данном пункте проводится описание поставленной задачи и возможных ограничений и т.п. Предметную область можно определить как объект или производственную систему со всем комплексом понятий и знаний о ее функционировании. При рассмотрении проблемной области необходимы знания о задачах, решаемых в производственной системе (сфере предполагаемого использования), и возможных ограничениях функционирования. Следует подробно описать проблематику решаемой задачи и по возможности привести UML диаграмму потоков данных. В случае обработки графических/видео и иных данных возможно приведение примеров исходных данных.

*Обзор основных систем* – проводится краткий обзор программных продуктов и систем, которые выполняют решение схожих задач или в которых реализованы алгоритмы необходимые для достижения поставленной цели. По итогам обзора делается небольшое сравнение по критериям (с представлением в табличной форме) или же в свободной форме на основе преимуществ/недостатков функционального характера. В конце раздела пишется краткий вывод на основе анализа функционала, организации и особенностей рассмотренных программных продуктов.

*Обзор технологий* – в данном пункте рассматриваются технологии необходимые для решения поставленной задачи (в том числе и организации защиты ПО). По возможности необходимо при описании технологий предоставить классификацию существующих методов для решения задачи в рамках рассмотренной технологии. Так возможно описание технологии с позиции выбранной среды разработки. В зависимости от предметной области возможно описание средств разработки (включая наборы компонентов и библиотек).

*Обзор методов и алгоритмов* – Приводится краткое описание группы основных методов и алгоритмов при помощи которых можно решить поставленную задачу. В зависимости от предметной области возможно описание вариантов использования рассмотренных библиотек/компонентов для реализации задачи. Помимо этого рассматриваются подходы к реализации системы плагинов с учетом выбранных языков программирования. В завершение рассматриваются алгоритмы необходимые для реализации механизма защиты, с учетом описанного и выбранного в пункте «Обзор технологий», подхода к организации защиты.

В конце главы приводятся выводы описывающие результат анализа технологий, методов и алгоритмов, пригодных для решения задач и указывается целесообразность выбора тех или иных технологий, методов и алгоритмов для проектирования программного продукта

## **5.2 Пояснение к содержанию пункта посвященного описанию структуры программного продукта**

Приводится подробное описание разработанного программного продукта с точки зрения его реализации. Так, в разделе приводится структура взаимодействия функциональных модулей программы, для модулей может быть отражена UML-диаграмма основных классов.

При наличии в программном обеспечении разветвленного меню приводится его структура, которая оформляется как рисунок (как правило, в виде иерархического дерева).

При описании каждого функционального модуля системы/программного продукта приводятся схемы описывающие процессы взаимодействия модуля с окружением (иными модулями/плагинами/источниками входных наборов данных). Указываются сведения о возможности многопоточной работы и т.п.

Также в этом разделе приводится схема взаимодействия с плагинами (динамически подключаемыми модулями, библиотеки функций *dll*) и описание интерфейсов взаимодействия с точки зрения программной реализации. При написании пункта «Организация системы плагинов» необходимо привести UML-диаграммы активности подключения плагинов, организации цепочки обработки, и формирования интерфейса настройки параметров алгоритма, реализованного в плагине.

## **5.2 Пояснение к содержанию пункта алгоритмической реализации**

В данном пункте необходимо привести описание основных алгоритмов реализуемых в программном продукте, с условным разбиением по функциональному назначению и/или реализуемым плагином. При описании алгоритмов необходимо привести схемы их функционирования и дать краткое пояснение. Для описания алгоритмов могут использоваться диаграммы активности, диаграммы потоков данных и/или блок-схемы. При описании алгоритмов реализации механизма защиты следует привести схемы хранения данных (описание класса/структуры) а также схему логику проверки.

## **5.3 Пояснение к содержанию пунктов содержащих руководства программиста и пользователя**

### **Руководство программиста**

Руководство программиста оформляется в соответствии с ГОСТ 19.504-79. Структура и оформление документа устанавливаются в соответствии с ГОСТ 19.105-78. Составление информационной части (аннотации и содержания) является обязательным.

Руководство программиста должно содержать следующие разделы: назначение и условия применения программы; характеристики программы; обращение к программе; входные и выходные данные; описание основных функций.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

В разделе "Назначение и условия применения программы" должны быть указаны назначение и функции, выполняемые программой; условия,

необходимые для выполнения программы (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению т.п.).

В разделе "Характеристики программы" должно быть приведено описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и самовосстанавливаемости программы и т.п.).

В разделе "Обращение к программе" должно быть приведено описание процедур вызова программы (способы передачи управления и параметров данных и др.).

В разделе "Входные и выходные данные" должно быть приведено описание организации используемой входной и выходной информации и, при необходимости, ее кодирования.

В разделе " описание основных функций " должно быть приведено описание модулей и хранящихся в них функций с пояснением назначений функций.

В приложении к руководству программиста могут быть приведены дополнительные материалы (примеры, иллюстрации, таблицы, графики и т.п.).

### **Краткое руководство пользователя**

Краткое руководство пользователя оформляется в соответствии с ГОСТ 19.505-79. Структура и оформление документа устанавливаются в соответствии с ГОСТ 19.105-78. Составление информационной части (аннотации и содержания) является обязательным.

Руководство должно содержать следующие разделы: назначение программы; условия выполнения программы; выполнение программы.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

В разделе "Назначение программы" должны быть указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе "Условия выполнения программы" должны быть указаны условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратных и программных средств и т.п.).

В разделе "Выполнение программы" должны быть указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Допускается содержание разделов иллюстрировать поясняющими примерами, таблицами, схемами, графиками. В приложения к руководству оператора допускается включать различные материалы, которые нецелесообразно включать в разделы руководства.

### **5.3 Пояснение к содержанию пункта тестирования разработанного программного продукта**

В данном разделе необходимо представить информацию о методиках тестирования и самом объекте тестирования, а также о результатах проведенного тестирования. Для различных объектов тестирования (программа в целом, отдельный алгоритм и т.п.) могут применяться различные тестовые наборы и методики, в связи с этим необходимо для каждого объекта тестирования привести описание методик тестирования и полученных результатов в табличном и/или графическом виде.

## **6. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ПРИЛОЖЕНИЯ «ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА»**

Техническое задание на разработку программного продукта выполняется на основе задания на курсовой проект, выданного преподавателем, и анализа деятельности организации, включающего следующие основные положения:

- выявление требований, предъявляемых к будущей системе;
- определение перечня целевых задач организации;
- определение потоков документооборота;
- анализ распределения функций по подразделениям и сотрудникам;
- определение перечня применяемых средств автоматизации.

В качестве исходных документов служит нормативно-справочная документация, перечень алгоритмов, научная литература, документы отчетности, приказы, распоряжения и т. п.

Краткое техническое задание согласовывается с руководителем курсовой работы через 1 – 2 недели после выдачи задания. Развернутое техническое задание, определяющее функционал разрабатываемого программного продукта, формируется и согласовывается с руководителем через 2 – 3 недели после выдачи задания.

Техническое задание оформляется в соответствии с ГОСТ 19.201-78. Он устанавливает процесс построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

Техническое задание должно содержать следующие разделы: введение; основания для разработки; назначение разработки; требования к программе или программному изделию; стадии и этапы разработки; в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

В разделе "Введение" указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

В разделе "Основания для разработки" должны быть указаны: документ (документы), на основании которых ведется разработка; организация, утвердившая этот документ, и дата его утверждения; наименование и (или) условное обозначение темы разработки.

В разделе "Назначение разработки" должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

Раздел "Требования к программе или программному изделию" должен содержать следующие подразделы: требования к функциональным характеристикам; требования к надежности; условия эксплуатации; требования к составу и параметрам технических средств; требования к информационной и программной совместимости; специальные требования.

В подразделе "Требования к функциональным характеристикам" должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.д. В подразделе "Требования к надежности" должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.).

В подразделе "Условия эксплуатации" должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т.п. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала. В подразделе "Требования к составу и параметрам технических средств" указывают необходимый состав технических средств с указанием их основных технических характеристик. В подразделе "Требования к информационной и программной совместимости" должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой. При необходимости должны обеспечиваться защита информации и программ.

В разделе "Стадии и этапы разработки" устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

В приложениях к техническому заданию, при необходимости, приводят: перечень научно-исследовательских и других работ, обосновывающих разработку; схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке; другие источники разработки.

## **7. ПРИМЕРНЫЕ ТЕМАТИКИ ДЛЯ КУРСОВОЙ РАБОТЫ**

Для выполнения курсовой работы можно выбрать **любую тему**, в том числе не относящуюся к нижеуказанному списку. **Главное чтобы в рамках курсовой работы была осуществлена реализация программного продукта с применением механизма модулей расширения.** Тему курсовой работы необходимо согласовать с преподавателем.

Ниже представлены тематика для определения будущей темы курсовой работы, формулировку которых можно взять за основу.

1. Разработка системы семантического анализа текстовых файлов.

2. Разработка программного комплекса для инвентаризации материальных ценностей организации.
3. Разработка приложения для пакетного транскодирования видео файлов на основе ffmpeg.
4. Разработка приложения расчета метрик качества для изображений и видеопоследовательностей
5. Система мониторинга сетевых устройств на основе протокола SNMP.
6. Система планирования маршрута курьера \*\*\*\*
7. Разработка приложения для извлечения субтитров из видеопотока с возможностью автоматического перевода.
8. Разработка аудиоредактора с возможностью применения спецэффектов.
9. Разработка системы расчёта расхода строительных материалов при постройке малоэтажных деревянных строений.
10. Разработка проигрывателя интернет-радио с настройкой схем визуализации.
11. Разработка системы анализа алгоритмов поиска особых точек на изображениях.
12. Разработка программного комплекса для исследования алгоритмов \*\*\*\*.
13. Разработка системы планирования грузоперевозок организации.
14. Разработка игрового тайм-менеджмент приложения.
15. Разработка системы локализации лиц на фотографиях и/или видеопоследовательностях.
16. Разработка программного комплекса агрегации информации о \*\*\*\*.
17. Разработка редактора web-документов с механизмом контроля версий.
18. Разработка системы формирования коллажа на основе набора фотографий/видео
19. Проектирование системы домашнего видеонаблюдения с возможностью улучшения визуального качества видео.
20. Разработка программного комплекса мониторинга за передвижением людей в помещении.
21. Разработка программной системы для автоматизации разработки мобильных приложений.
22. Разработка системы для трехмерного моделирования дизайна помещения.
23. Разработка программного комплекса исследования алгоритмов шумоподавления || для аудио/изображений/видео
24. Проектирование системы матричных вычислений.
25. Разработка системы анализа информационных порталов с возможностью сохранения документов в базе данных.
26. Разработка системы визуальной генерации SQL-кода для работы с СУБД \*\*\*\*
27. Разработка рекомендательной системы по выбору ноутбуков/смартфонов на основе агрегации результатов \*\*\*\*
28. Разработка программного комплекса для проведения тестирования \*\*\*
29. Проектирование системы анализа интернет ресурсов для \*\*\*\*
30. Разработка системы редактирования фотографий с возможностью применения спецэффектов.

31. Разработка программного комплекса для визуальной настройки конфигурации нейронных сетей для \*\*\*\*
32. Разработка приложения для автоматизированной генерации РНР кода «Конструктор сайтов»
33. Разработка виртуального лабораторного комплекса для проведения занятий по \*\*\*\*
34. Разработка программного комплекса автоматизации проведения олимпиады по программированию.
35. Разработка редактора субтитров с поддержкой настройки стилей.
36. Разработка программного комплекса для автоматизации исследования параллельных реализаций алгоритмов обработки изображений/видео\*\*\*
37. Разработка приложения для планирования закупок материалов для изготовления корпусной мебели.
38. Разработка система автоматизированной генерации кода скриптов Avisynth.
39. Разработка программного комплекса для контрольно-пропускного пункта \*\*\*\*
40. Разработка приложения для автоматического формирования коллажей с использованием галереи облачного хранилища.
41. Разработка программной системы видеонаблюдения за сотрудниками.
42. Разработка аналитической системы для мониторинга динамики \*\*\*\*
43. Разработка приложения для анализа данных \*\*\*\*
44. Разработка веб-приложения для обработки изображений.
45. Разработка мобильного приложения для поиска объявлений \*\*\*\* на \*\* платформах

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет науки и технологий  
имени академика М.Ф. Решетнева»

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**КУРСОВАЯ РАБОТА**

Конструирование программного обеспечения

Первая строка названия темы

---

вторая строка названия темы

---

Руководитель \_\_\_\_\_

\_\_\_\_\_

А.Г. Зотин

инициалы, фамилия

Обучающийся БПИ\*\*-0\*, \*\*1214\*\*\*

номер группы, зачетной книжки

\_\_\_\_\_

И.И. Иванов

инициалы, фамилия

Красноярск 20\*\* г.

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ЗАДАНИЕ**

на курсовую работу по дисциплине Конструирование программного обеспечения  
обучающемуся Иванову Ивану Ивановичу

Группа БПИ \*\*-0\* Форма обучения очная

Тема работы: Первая строка названия темы  
вторая строка названия темы

Срок сдачи курсовой работы \*\* .02.202\*

Перечень вопросов, подлежащих разработке при написании теоретической части:

---

---

---

---

---

---

---

---

---

---

Перечень вопросов, подлежащих разработке при написании практической части:

---

---

---

---

---

---

---

---

---

---

Дата выдачи задания: \*\* .06.202\*

Руководитель Зотин А.Г., доцент кафедры ИВТ  
(подпись)

Задание принял к исполнению (дата) \_\_\_\_\_

\_\_\_\_\_  
(подпись обучающегося, И.О. Фамилия)

## Пример структуры и оформления введения курсовой работы

### ВВЕДЕНИЕ

*Актуальность.* На сегодняшний день ... .

...

Таким образом, ... является актуальной задачей.

*Цель и задачи.* Целью курсовой работы является ... .

!! Целью курсовой работы нельзя ставить разработку программного обеспечения. Целью будет то, для чего нужна программа т.е. автоматизировать/упростить что-либо.

Для достижения поставленной цели необходимо решить следующие задачи:

- выполнить анализ \*\*\* (согласно предметной области);
- определить \*\*\* автоматизации и группы пользователей;
- осуществить обзор программного обеспечения в рассматриваемой предметной области;
- .....
- .....
- .....
- .....
- .....
- .....
- .....
- осуществить программную реализацию ...;
- провести тестирование разработанной \*\*\*.

*Структура работы.* Пояснительная записка к курсовой работе состоит из введения, двух глав, заключения и списка использованных источников из ... наименований. Изложена на ... страницах и содержит ... рисунков и ... таблиц.

В первой главе курсовой работы приводится ... , отражен ... .

Во второй главе ... .

В заключении ... .

!! При описании структуры работы ориентироваться на содержание пояснительной записки, писать фразы с позиции констатации наличия материала на «бумаге». Например: приводится ...; показаны ...; отражено ...; содержится ...; представлено(ы) ....

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ

### Введение

В настоящее время ... . В связи с этим разработка ... .

### Общие положения.

Полное наименование: \*\*\*\*\*.

Краткое наименование: \*\*\*\*\*.

Вид разработки: \*\*\*\*\*.

Разработчик: \*\*\*\*\*.

### Основание для разработки

Разработка программного продукта ведется на основании учебного плана СибГУ им. М.Ф. Решетнева по направлению подготовки 09.03.04 Программная инженерия, направленности «Разработка программно-информационных систем».

### Плановые сроки начала и окончания выполнения работы

Плановый срок начала работ – \*\*.\*\*.\*\*.20\*\*

Плановый срок окончания работ – \*\*.\*\*.\*\*.20\*\*

### Назначение разработки

Разрабатываемый программный продукт предназначен для ... .

### Требования к программе

#### *Требования к функциональным характеристикам*

В данном блоке раскрываются функциональные требования, ниже приведены пример фрагментов написания требований. Также можно требования писать обычным текстом, но в таком ключе, чтобы было понятно какие функциональные возможности будут у разработки. Ниже приведены примеры возможных фраз для формирования функциональных требований.

В программе должно быть предусмотрено ...

Программа должна:

обеспечить пользователю возможность ... ;

– позволять ... ;

– осуществлять ... ;

– ... ;

– ... .

Помимо этого для соответствия указанному назначению, состав выполняемых функций должен включать следующее:

– формирование ... ;

– генерация ... ;

– ... .

Ввод данных должен выполняться в программе ... . Вывод информации должен выполняться в виде ... .

Создаваемые отчеты в **программе** должны предоставлять следующую информацию:

- ... ;
- ... ;
- ... .

*Требования к надежности*

В **программе** должен быть обеспечен контроль за ... . Возникновение каких-либо внутренних ошибок не должно приводить к утере данных. **Программа** должна быть ... .

Для защиты от несанкционированного использования должно быть предусмотрено ...

*Условия эксплуатации*

Приложение рассчитано на эксплуатацию пользователями ... .

*Требования к составу и параметрам технических средств*

- процессор \*\* с частотой не ниже ... ГГц.;
- оперативная память не меньше ... Гб.;
- не менее ... Гб свободного места на жестком диске;
- ... .

*Требования к информационной и программной совместимости*

- операционная система Windows 10/11;
- ... .

**Состав и содержание работ по выполнению курсовой работы**

Таблица А.1 – Календарный план-график выполнения стадий и этапов разработки

Наименование работ	Сроки выполнения
Обзор/Анализ *** предметной области	
Обзор ... программного обеспечения ...	
***	
***	
***	
Разработка механизма модулей расширения (плагинов)	
***	
Разработка механизма защиты	
Разработка пользовательского интерфейса	
Программная реализация ***	
***	
Формирование руководств программиста и пользователя	
Тестирование и отладка ***	
Создание инсталляционного комплекта	
Оформление пояснительной записки	
Предоставление преподавателю всех необходимых для защиты материалов	** .06.20**