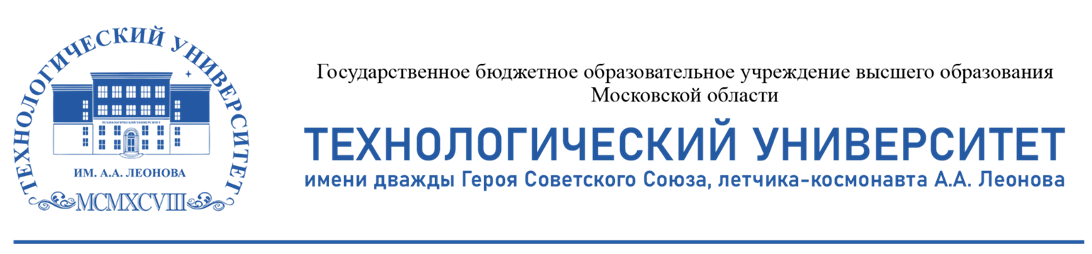
****Колледж космического машиностроения и технологии**

**ОТЧЕТ**

по производственной практике ПП.03.01

по профессиональному модулю ПМ.03 «Участие в интеграции программных модулей»

специальность 09.02.03 «Программирование в компьютерных системах»

Выполнил студент

\_\_\_\_\_\_\_\_\_\_

группы: П1-18

Проверил преподаватель

Бобкова Н. Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2021

**Примерное содержание отчета по ПП.03**

[Введение 1](#_Toc68814168)

[1. Задание на производственную практику 1](#_Toc68814169)

[2. Общие сведения о предприятии 2](#_Toc68814170)

[3. Проектирование 2](#_Toc68814171)

[4. Технико-экономические показатели 2](#_Toc68814172)

[Заключение 3](#_Toc68814173)

[Дневник производственной практики на предприятии «ККМТ» 3](#_Toc68814174)

[Дневник производственной практики на предприятии 3](#_Toc68814175)

[Приложение 1. Руководство оператора 3](#_Toc68814176)

[Приложение 2. Руководство программиста 3](#_Toc68814177)

[Приложение 3. Листинг кода 3](#_Toc68814178)

# 

# Введение

При обучении на 4 курсе МГОТУ “Колледж космического машиностроения и технологий”, студентами группы П1-18 Волковым Максимом, Денисовым Матвеем и Курбанмухаммедовым Стивом, Ларченко Михаилом была пройдена практика по модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем».

# Задание на производственную практику

1. *Предоставить общие сведения о предприятии.*
   1. *Сферу деятельности предприятия, история появления и развития предприятия.*
   2. *Структуру предприятия, функции подразделения и взаимосвязь с другими подразделениями.*
   3. *Список задач подразделения в сфере информационных технологий.*

# Содержание выполняемых видов работ

## 2.1 Анализ существующих в подразделении локальных и распределенных баз данных, приложений баз данных

На данный момент в подразделении отсутствуют базы данных, приложения базы данных.

## 2.2 Получение задания на конкретную разработку

Задание состоит в проектировании и создании программы, которая будет помогать проверять задания студентов.

Необходимо разработать настольное приложение для проверки заданий. Программа будет состоять из интерфейса, который упрощает поиск студентов и заданий. В приложении можно добавлять, удалять и изменять задаваемые параметры в колонке студентов и заданий.

Инструменты для реализации клиентской части: Free Pascal + Lazarus

Инструменты для реализации серверной части: SQLite

## 2.3 Изучение документооборота, подлежащего автоматизации

Документооборот в отделе формирования и реализации дистанционных технологий представляет из себя настольное приложение для проверки заданий, а по окончанию проверки заданий, выводит отчет обо всех студентов в группе.

Рисунок 2.

Документооборот



## 2.4 Построение IDEF-модели

Рисунок 3.

IDEF-модель 1-й уровень

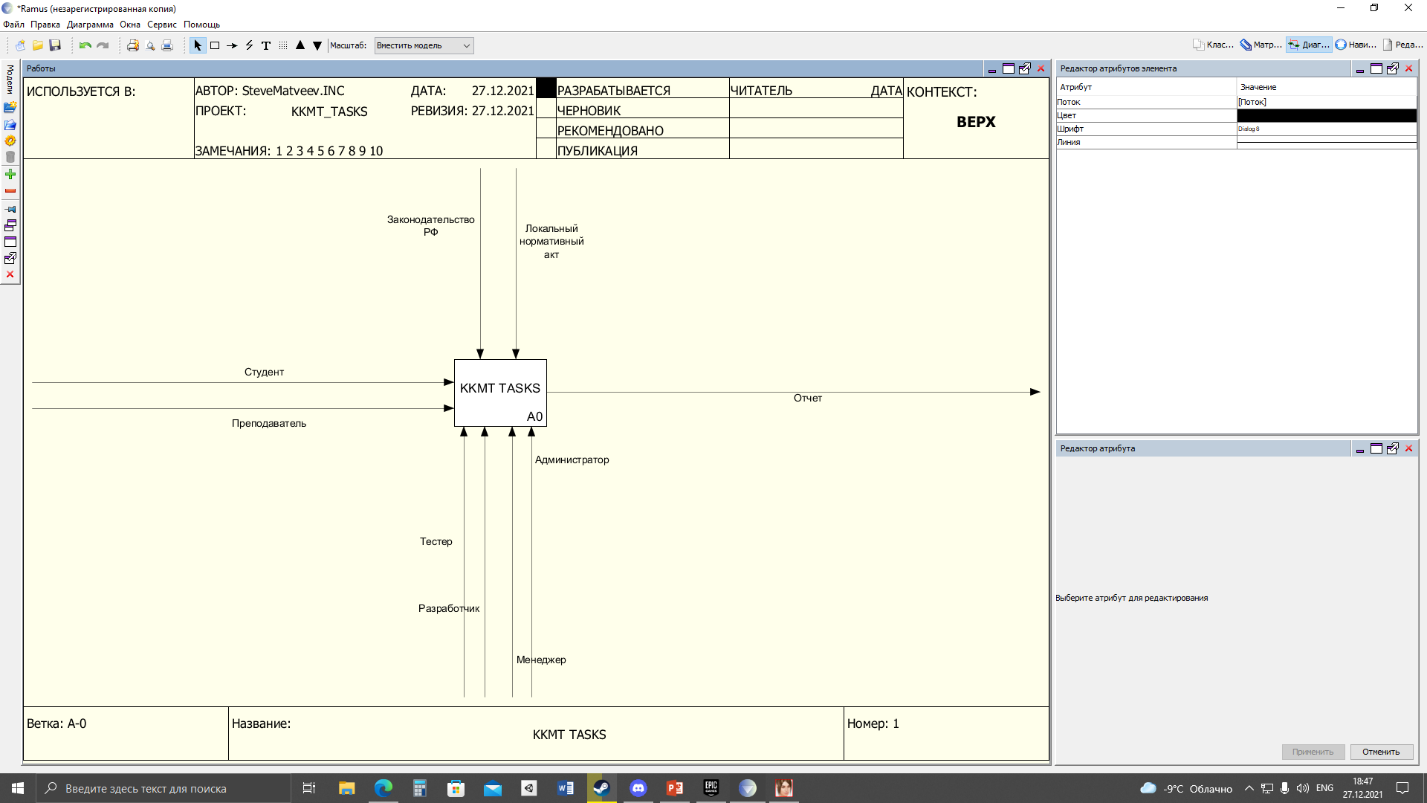
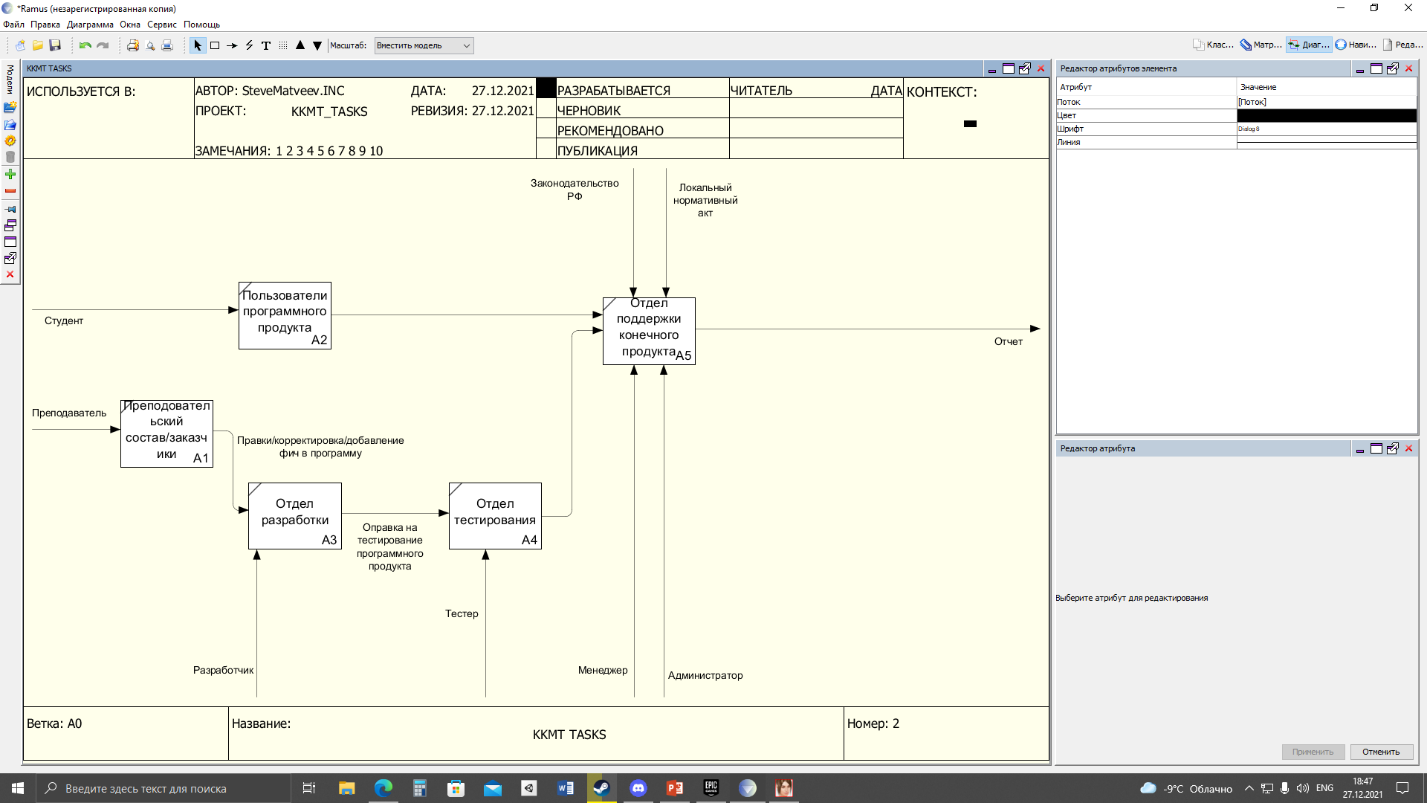


Рисунок 4

IDEF-модель 2-й уровень



* 1. *Составить техническое задание на разработку программного средства.*

1. **Введение**
   1. **Наименование программы**

Наименование программы – “KKMT-task 1.0.”.

* 1. **Краткая характеристика области применения**

1. Необходимо разработать настольное приложение для проверки заданий. Программа будет состоять из интерфейса, который упрощает поиск студентов и заданий. В приложении можно добавлять, удалять и изменять задаваемые параметры в колонке студентов и заданий.
2. **Основания для разработки**

Облегчение проверки заданий, преподавателями.

1. **Назначение разработки**
   1. **Функциональное назначение**

Добавление группы, добавление задания, редактирование и удаление задания, редактирование и удаление группы.

* 1. **Эксплуатационное назначение**

Программа будет эксплуатироваться на устройстве, которое использует ОС Windows 7 и выше. Ограничения к категориям пользователей отсутствуют.

1. **Требования к программе или программному изделию**
   1. **Требования к функциональным характеристикам**
      1. **Требования к составу выполняемых функции**

Программа должна содержать 1 окно. Главный экран, который содержит кнопки с верху для работы с фалом. **В разработке.**

* + 1. **Требования к организации входных и выходных данных**

Входные данные представляют собой название и содержание текстового файла.

* + 1. **Требования к временным характеристикам**

Все операции в программе не должны блокировать графический интерфейс более чем на 1337мс.

**Требования к надежности**

* + 1. **Требования к обеспечению надежного (устойчивого) функционирования программы**

Программа не должна аварийно завершаться при условии наличия достаточных ресурсов на устройстве (ОЗУ, Темп. ЦП и т.д.)

* + 1. **Время восстановления после отказа**

Время восстановления после отказа, вызванным критическим состоянием устройства не должно превышать 5 минут.

* + 1. **Отказы из-за некорректных действий оператора**

При некорректных действиях оператора программа не завершиться аварийно.

* 1. **Условия эксплуатации**

Программа должна запускаться как отдельная программа для ОС Windows 7 и выше, где установлен интерпретатор С#.

* + 1. **Климатические условия эксплуатации**

Специальные условия не требуются.

* + 1. **Требования к видам обслуживания**

Обслуживание не требуется.

* + 1. **Требования к численности и квалификации персонала**

Программа представляет собой пользовательское приложение, соответственно, персонал не требуется.

* 1. **Требования к составу и параметрам технических средств.**

Минимальный состав технических средств:

* ОС: Windows 7 и выше
* ОЗУ: 1 Гб
* ЦП: 1 ГГц.
* ПЗУ: 50 Мб.
  1. **Требования к информационной и программной совместимости**

Требования отсутствуют.

* 1. **Требования к маркировке и упаковке**

Программа просто переносится в виде папки, которая содержит необходимые файлы для работы программы.

* 1. **Требования к транспортированию и хранению**
  2. **Специальные требования**

Специальных требований нет.

1. **Требования к программной документации.**

Состав программной документации состоит только из технического задания.

1. **Технико-экономические показатели**

В разные операционные системы уже как правило встроен текстовый редактор, так что данная программа не будет востребована. Программа сделана для получения опыта в данной сфере.

1. **Стадии и этапы разработки**

Разработка состоит из следующих стадий:

* Разработка требований
* Проектирование
* Реализация
* Тестирование
* Внедрение

1. **Порядок контроля и приемки**

После разработки всей программы происходит финальное тестирование всего функционала группой контроля качества.

* 1. *Разработать модули программного средства в соответствии с ТЗ.*

## Описание главного модуля.

Главный модуль состоит из главной функции, которая запускается при старте программы инициализирует класс Application и создает главную форму.

Листинг 1 Главная функция

program KkmtTesting;

{$mode objfpc}{$H+}

uses

{$IFDEF UNIX}

cthreads,

{$ENDIF}

{$IFDEF HASAMIGA}

athreads,

{$ENDIF}

Interfaces, // this includes the LCL widgetset

Forms, MainFormUnit, GroupFormUnit, taskformUnit, SettingsFormUnit;

{$R \*.res}

begin

RequireDerivedFormResource:=True;

Application.Scaled:=True;

Application.Initialize;

Application.CreateForm(TMainForm, MainForm);

Application.Run;

end.

## Описание спецификаций к модулям

Всего разработано 6 модулей:

* **MainFormUnit**
* **SettingsFormUnit**
* **TaskFormUnit**
* **GroupFormUnit**
* **AppDatabase**
* **Utils**

## Описание модулей

**MainFormUnit** – Модуль, который содержит класс главной формы программы.

Рассмотрим функцию запуска обновления выполненных веток и построим блок схему:

Листинг 2. Функция обновления

procedure TMainForm.UpdateStatusButtonClick(Sender: TObject);

begin

if not NowUpdating then

begin

if (SelectedGroup = -1) and (SelectedTask = -1) then exit;

Label2.Caption := StaticText2.Caption;

Label4.Caption := StaticText1.Caption;

Thread := TReportUpdateThread.Create(True, SelectedGroup, SelectedTask);

Thread.Start;

end

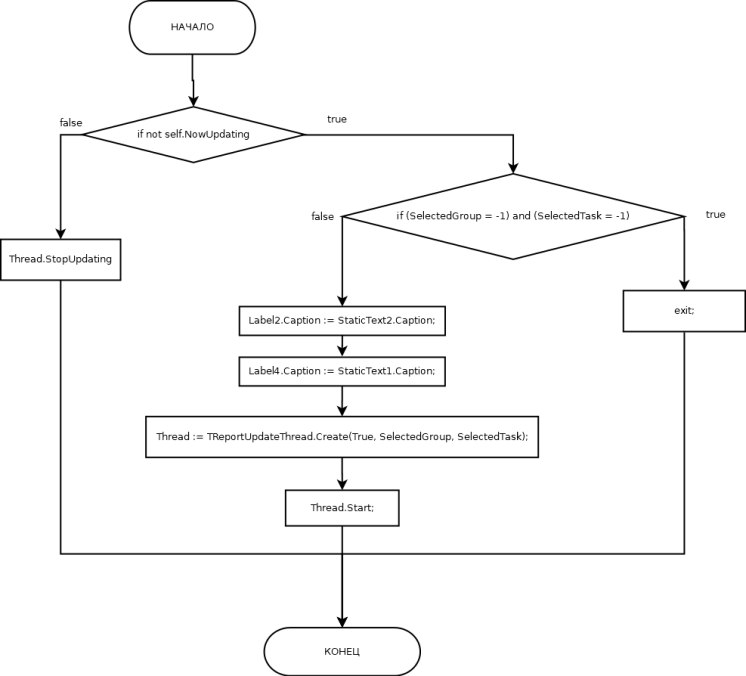
else

begin

Thread.StopUpdating;

end;

end;

Рисунок 7. Блок схема класса MainFormUnit

**SettingsFormUnit** - Модуль с классом формы настроек программы.

Имеет два текстовых поля:

* Логин пользователя GitHub
* Персональный токен для этого логина пользователя

При нажатии кнопки "Сохранить" записывает данные в поля в файл config.json в папке программы.

**TaskFormUnit** - Модуль с классом формы для добавления, изменения заданий.

Имеет поле ввода имени задания, а также дерево с ожидаемой структурой ветки начиная от корня ветки.

Также имеет три кнопки:

* Добавить элемент дерева как ребенка выделенного элемента
* Удалить выделенный элемент
* Изменить название выделенного элемента

**GroupFormUnit** - Модуль с классом формы для редактирования, добавления групп и их репозитория для проверки.

Имеет два поля:

* Имя группы
* Уникальный идентификатор репозитория группы

При сохранении данных этой формы выполняется проверка на существование репозитория.

**AppDatabase** - Модуль содержащий классы и функции для работы с базой данных приложения.

Имеет в себе класс TAppDatabase с методами для выполнения запросов с параметрами.

* Exec - для выполнения простого SQL запроса
* ExecOut - для выполнения SQL запроса с возвращением результата

**Utils** - Хранит в себе функции различного назначения.

* CreateGitHubHTTP - для инициализации объекта TFPHttpClient для работы с GitHub API
* OpenDialog - для открытия формы в виде простого диалога

## Описание тестовых наборов модулей

В некоторых модулях производится проверка данных на корректность, чтобы сообщить пользователю о проблеме и предотвратить аварийное завершение программы.

В GroupFormUnit происходит проверка на существование репозитория при нажатии на кнопку "Button1".

**Тест 1. Проверка имя группы**

Проверка происходит на пустоту имени группы и на пробелы. При непрохождении проверки выводится messagebox с сообщением "Имя группы не должно быть пустым или иметь пробелы".

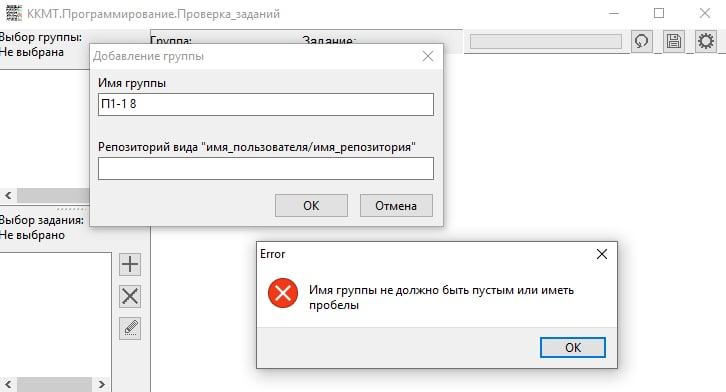


Рисунок 8. Ошибка при не правильном вводе группы

**Тест 2. Проверка заполнения вида репозитория**

Проверка на существование репозитория. Если он не существует, то будет показана ошибка "Репозиторий не найден".

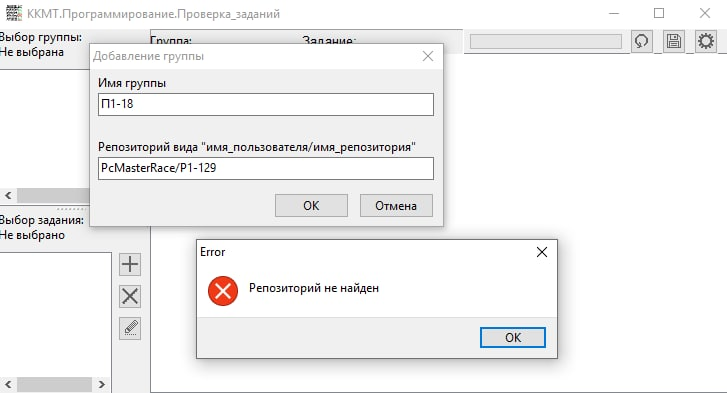
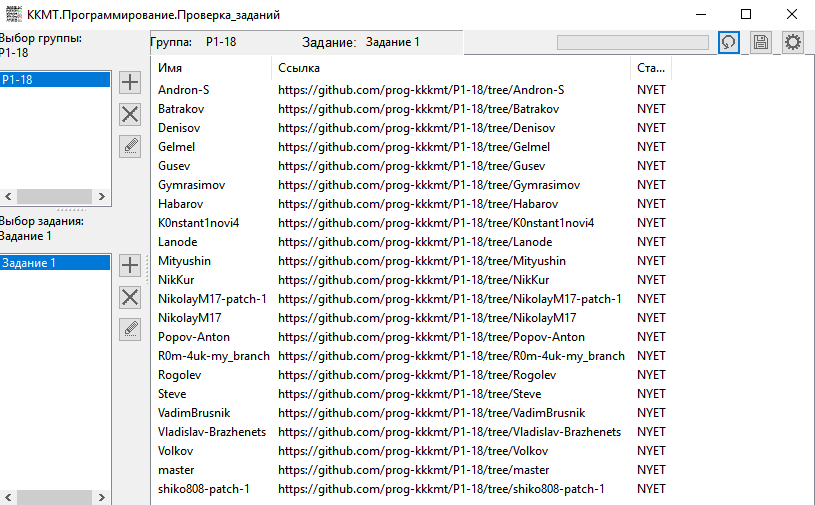


Рисунок 9. Ошибка при неправильном вводе репозитория

* 1. *Разработать пользовательский интерфейс программного средства.*
  2. *Провести отладку и тестирование программного средства.*

Во время разработки программы мы столкнулись с багом при котором не было проверки репозитория на существование. Поэтому при запуске обновления списка выполнения задания программа выдавала ошибку и не могла продолжить работу дальше.

После того как мы добавили проверку на наличие репозитория в диалог добавления новой группы, баг исчез



* 1. *Разработать комплект технологической документации.*

1. *Сделать выводы по результатам практики.*

# 2. Общие сведения о предприятии

2.1. Сфера деятельности предприятия, история появления и развития предприятия *(Сферой деятельности предприятия «Наименование предприятия» является…, краткая история развития предприятия).*

2.2. Структура предприятия, функции подразделения и взаимосвязь с другими подразделениями *(описание подразделений предприятия, можно представить схематично или включить скан структуры предприятия).*

2.3. Список задач подразделения в сфере информационных технологий *(описать деятельность подразделения информационных технологий, например,* *реализация концепции разработки, обеспечение исполнения планов и графиков работ по разработке, внедрение и сопровождение ИТ-сервисов, администрирование, ввод в эксплуатацию и техническая поддержка, совершенствование системы управления предприятием за счет применения современных информационных систем, выявление резервов по повышению эффективности ИТ-сервисов предприятия и т.д.).*

# 3. Проектирование

3.1. Анализ существующих в подразделении локальных и распределенных баз данных, приложений баз данных *(описать используемое на предприятии ПО).*

3.2. Задание на конкретную разработку *(описать будущую разработку).*

3.3. Изучение документооборота, подлежащего автоматизации *(описать применяемый документооборот предприятия, выявить недостатки, достоинства, провести анализ возможностей улучшения работы за счет внедрения Вашего будущего ПС)*.

3.4. Построение IDEF-модели *(описании бизнес-процессов с помощью кейс-средства Ramus, приводим в отчете скрины).*

3.5. Составление технического задания на разработку программного средства.

3.6. Разработка модулей программного средства в соответствии с ТЗ.

3.7. Разработка пользовательского интерфейса программного средства.

3.8. Отладка и тестирование программного средства.

# Заключение

# Дневник производственной практики на предприятии «ККМТ»

# Дневник производственной практики на предприятии

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работы | Подпись руководителя практики от предприятия |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Руководитель: \_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_

Приложение 1. Руководство оператора.

Приложение 2. Руководство программиста**.**

Приложение 3. Листинг кода**.**