

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Энергетический

Направление подготовки 13.04.02 Электроэнергетика и электротехника

Кафедра Электроэнергетический систем и электротехники

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Создание графического интерфейса для программы дорасчета телеизмерений и телесигнализации

УДК 621.397.083.7:004.514

Студент

Группа	ФИО	Подпись	Дата
5AM5Д	Симонов Игорь Викторович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Васильев Алексей Сергеевич	К.Т.Н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры менеджмента	Фигурко А.А.	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Невский Е.С.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ЭСиЭ	Прохоров А.В.	К.Т.Н.		

Томск – 2017 г.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Энергетический
Направление подготовки 13.04.02 Электроэнергетика и электротехника
Кафедра Электроэнергетический систем и электротехники

УТВЕРЖДАЮ:
Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской выпускной квалификационной работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
5АМ5Д	Симонову Игорю Викторовичу

Тема работы:

Создание графического интерфейса для программы дорасчета телеизмерений и телесигнализации

Утверждена приказом директора (дата, номер)

08.02.2017, № 626/с

Срок сдачи студентом выполненной работы:

01.06.2017

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

1. Оперативно-информационный комплекс СК-2007;
2. Руководство по эксплуатации оперативно-информационного комплекса СК-2007;
3. Объект исследования – графический интерфейс проектирования алгоритмов дорасчета телеизмерений и телесигнализации.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Анализ литературы по теме способов и методов отображения и графического проектирования вычислительных алгоритмов; 2. Обзор современных программных комплексов, использующих графические средства проектирования алгоритмов; 3. Разработка интерфейса графического проектирования алгоритмов дорасчета телеизмерений и телесигнализации, включая анализ подлежащих автоматизации технологических задач и формирование требований к графическому интерфейсу.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Блок-схемы алгоритмов чтения и записи XML-файла, содержащего информацию о выполняемом алгоритме дорасчета и описание его графического отображения.</p>

<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>1. Анализ применимости пользовательских интерфейсов</p>	<p>Васильев А.С.</p>
<p>2. Функциональные требования к графическому редактору</p>	<p>Васильев А.С.</p>
<p>3. Разработка графического редактора</p>	<p>Васильев А.С.</p>
<p>4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Фигурко А. А.</p>
<p>5. Социальная ответственность</p>	<p>Невский Е.С.</p>
<p>6. Creating a graphical interface for the program of calculating telemetry and telesignaling</p>	<p>Исакова Ю.И.</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Разделы 1-5 – Русский язык</p>	
<p>Раздел 6 – Английский язык</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	
--	--

Задание выдал руководитель:

<p>Должность</p>	<p>ФИО</p>	<p>Ученая степень, звание</p>	<p>Подпись</p>	<p>Дата</p>
<p>Старший преподаватель</p>	<p>Васильев Алексей Сергеевич</p>	<p>к.т.н.</p>		

Задание принял к исполнению студент:

<p>Группа</p>	<p>ФИО</p>	<p>Подпись</p>	<p>Дата</p>
<p>5AM5Д</p>	<p>Симонов Игорь Викторович</p>		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
Профессиональные компетенции	
Р1	Применять соответствующие гуманитарные, социально-экономические, математические, естественно-научные и инженерные знания, компьютерные технологии для решения задач расчета и анализа электрических устройств, объектов и систем.
Р2	Уметь формулировать задачи в области электроэнергетики и электротехники, анализировать и решать их с использованием всех требуемых и доступных ресурсов.
Р3	Уметь проектировать электроэнергетические и электротехнические системы и их компоненты.
Р4	Уметь планировать и проводить необходимые экспериментальные исследования, связанные с определением параметров, характеристик и состояния электрооборудования, объектов и систем электроэнергетики и электротехники, интерпретировать данные и делать выводы.
Р5	Применять современные методы и инструменты практической инженерной деятельности при решении задач в области электроэнергетики и электротехники.
Р6	Иметь практические знания принципов и технологий электроэнергетической и электротехнической отраслей, связанных с особенностью проблем, объектов и видов профессиональной деятельности профиля подготовки на предприятиях и в организациях – потенциальных работодателях.
Универсальные компетенции	
Р7	Использовать знания в области менеджмента для управления комплексной инженерной деятельностью в области электроэнергетики и электротехники
Р8	Использовать навыки устной, письменной речи, в том числе на иностранном языке, компьютерные технологии для коммуникации, презентации, составления отчетов и обмена технической информацией в областях электроэнергетики и электротехники.
Р9	Эффективно работать индивидуально и в качестве члена или лидера команды, в том числе междисциплинарной, в области электроэнергетики и электротехники.

P10	Проявлять личную ответственность и приверженность нормам профессиональной этики и нормам ведения комплексной инженерной деятельности.
P11	Осуществлять комплексную инженерную деятельность в области электроэнергетики и электротехники с учетом правовых и культурных аспектов, вопросов охраны здоровья и безопасности жизнедеятельности.
P12	Быть заинтересованным в непрерывном обучении и совершенствовании своих знаний и качеств в области электроэнергетики и электротехники.

Реферат

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. Актуальность проблемы определяется ограниченностью штатных средств ОИК СК-2007 создания формул оперативного дорасчета в плане реализации сложных вычислительных алгоритмов и отсутствием понятного графического редактора, который способствует наглядному отображению алгоритма формул оперативного дорасчета.

Цель и задачи исследования. Целью настоящей работы является разработка программного модуля, позволяющего создавать формулы дорасчета в графическом виде при помощи функциональных блоков.

Задачи:

1. Провести анализ существующих графических алгоритмов и языков графического программирования;
2. Сформировать основные и второстепенные требования к программному модулю;
3. Разработать алгоритм работы и интерфейс программы с учетом требований нормативно - технической документации.

Методологическая основа. При работе над диссертацией использовались основные положения нормативной документации о графическом программировании, эргономике пользовательских интерфейсов. Большинство выводов работы основано на результатах исследований автора.

Информационная база. Поставленная цель достигается в процессе изучения научной и специальной литературы, интернет - источников; собственных исследований.

Новизна исследования и научные результаты. Предложен способ визуализации алгоритма с помощью функциональных блоков,

ориентированный на технологические задачи оперативно-диспетчерского управления режимами электроэнергетических систем.

Объектом исследования является графический интерфейс проектирования алгоритмов дорасчета телеизмерений и телесигнализации.

Предметом исследования является способы и методы визуализации алгоритмов дорасчета телеизмерений и телесигнализации, а также процесс разработки специализированных программ, использующих графические методы отображения.

Апробация работы: публикация материалов в сборнике VII международной научно-технической конференции «Электроэнергетика глазами молодежи – 2016» (г. Казань, КГЭУ, 19-23 сентября 2016 года).

Положения, выносимые на защиту:

1. Виды и способы графического программирования;
2. Требования к графическим интерфейсам;
3. Алгоритм работы и интерфейс программы с учетом требований нормативно - технической документации.

Практическая значимость. Ожидаемым эффектом от внедрения разработанной графической среды программирования является повышение эффективности разработки и реализации сложных вычислительных алгоритмов, выполняемых в программе дорасчета телеизмерений и телесигнализации для задач оперативно-информационного комплекса СК-2007 эксплуатируемого в АО «СО ЕЭС».

Структура работы. Работа состоит из введения, пяти глав, заключения, списка литературы, двух приложений. Текстовая часть проиллюстрирована 14 рисунками, список источников включает такие нормативно - правовые документы как: ГОСТ Р МЭК 61131-3-2016 Контроллеры программируемые.

Часть 3. Языки программирования; ИЕС 61499-1 Function blocks – Part 1:
Architecture - Edition 2.0.

ОСНОВНОЕ СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Глава 1. Рассматриваются основные виды графического интерфейса, языки графического программирования и приложения, которые имеют схожий подход к формированию алгоритмов с выбранным языком программирования. Формируются общие требования к программному модулю на основании изложенной в данной главе информации, а также с помощью сравнительного анализа имеющихся графических интерфейсов.

Глава 2. Посвящена обзору основных и второстепенных задач программного модуля, определению целевой аудитории и взаимодействию со сторонними приложениями с целью формирования программных требований к разрабатываемому модулю. Обоснован выбор среды разработки и языка программирования.

Глава 3. Раскрывает вопрос разработки алгоритма работы программы в среде программирования Microsoft Visual Studio на языке программирования C#. В данной главе показаны основные компоненты графического интерфейса, реализация подключения к базе данных реального времени, формирование конфигурационного файла для программы дорасчета телеизмерений и телесигнализации.

Глава 4. Посвящена вопросам финансового менеджмента, ресурсоэффективности и ресурсосбережения применительно к разработке программного модуля «Графический редактор». Выполнен предпроектный анализ, инициация проекта, планирование управления научно-техническим проектом, расчет сметы затрат на исследование.

Глава 5. Рассматриваются вопросы социальной ответственности, к которым относится анализ основных причин нарушения работы программного модуля рассмотрение угроз при возникновении сбоев работы программы, избежание негативных последствий при некорректной работе алгоритмов программного модуля.

Определения, обозначения, сокращения, нормативные ссылки

В настоящей работе использованы ссылки на следующие стандарты:

IEC 61499-1

ГОСТ Р МЭК 61131-3-2016

ГОСТ Р ИСО/МЭК 9126-93

В работе использованы следующие обозначения и сокращения:

АО «СО ЕЭС» - акционерное общество «Системный оператор Единой энергетической системы»;

ОИК – оперативно-информационный комплекс;

РЗ – релейная защита;

ПА – противоаварийная автоматика;

НСИ – нормативно-справочная информация;

ТИ – телеизмерения;

ТС – телесигнализация;

ОС – операционная система;

ПЛК – программируемый логический контроллер;

ПО – программное обеспечение;

БД – база данных;

КДД – компонент доступа к данным;

SQL - structured query language (язык структурированных запросов);

XML - extensible markup language) — расширяемый язык разметки.

Оглавление

Реферат	6
Определения, обозначения, сокращения, нормативные ссылки	10
Введение	13
1 Анализ применимости пользовательских интерфейсов	15
1.1 Принципы построения пользовательских интерфейсов	15
1.2 Виды пользовательского интерфейса	21
1.2.1 Описание графического интерфейса	22
1.2.2 Описание командного интерфейса	23
1.2.3 Сравнительный анализ командного и графического интерфейсов 24	
1.3 Анализ графических языков программирования	26
1.3.1 Виды графических языков программирования	26
1.3.2 Пример реализации графических языков программирования	29
1.4 Общие требования к графическому редактору	32
1.5 Вывод по 1 главе	32
2 Функциональные требования к графическому редактору	34
2.1 Определение целевой аудитории графического редактора	34
2.2 Основные и второстепенные задачи	34
2.3 Взаимодействие с базами данных и программой дорасчета	36
2.4 Выбор языка и среды программирования	36
2.4.1 Анализ языков программирования	36
2.4.2 Среда разработки	39
2.5 Вывод по 2 главе	40
3 Разработка графического редактора	42
3.1 Общая структура графического редактора	42
3.2 Реализация создания, перемещения, соединения и удаления графических блоков	43
3.3 Считывание НСИ ОИК СК-2007	46
3.4 Запись/чтение XML-файла	47
3.5 Пример реализации алгоритма «Счетчик»	48

3.6	Вывод по 3 главе.....	50
4	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	52
4.1	Предпроектный анализ	52
4.1.1	Анализ конкурентных технических решений и потенциальные потребители результатов исследования	52
4.2	Инициация проекта	53
4.3	Планирование управления научно-техническим проектом	54
4.3.1	Иерархическая структура работ проекта	54
4.3.2	Организация и планирование работ	55
4.4	Расчет сметы затрат на исследование	57
4.5	Вывод по 4 главе.....	62
5	Социальная ответственность	64
5.1	Этапы разработки программы.....	64
5.2	Причины, приводящие к неправильному функционированию ОИК СК-2007.....	66
5.3	Причины, приводящие к неправильному функционированию графического интерфейса	67
5.4	Вывод по 5 главе.....	70
	Заключение	71
	Список используемых источников.....	72
	Приложение А	74
	Приложение Б.....	75

ВВЕДЕНИЕ

АО «СО ЕЭС» (Акционерное общество «Системный оператор Единой энергетической системы») России осуществляет централизованное оперативно-диспетчерское управление, а также обеспечивает перспективное развитие ЕЭС России. Для достижения необходимой надежности оперативного управления режимами энергосистем, диспетчерам должны быть доступны все необходимые данные, предоставляемые оперативно-информационным комплексом (ОИК) СК-2007 в виде таблиц и графиков на формах отображения. Данные, поступающие с устройств телемеханики, не являются достаточными для мониторинга и контроля, и приходится производить дорасчет недостающих параметров. Задание алгоритмов, по которым производится расчет требуемых величин, осуществляется в Мастере формул ОИК посредством текстового описания формул.

В процессе работы возникают случаи, когда необходимо реализовать сложный алгоритм, который находится вне функциональных возможностей модуль оперативного дорасчета ОИК. В связи с этим, используются программные модули, созданные специально под ту или иную технологическую задачу. Одним из таких модулей, который предназначен для реализации функций оповещения диспетчеров о срабатывании РЗ и ПА, является APTSLogic.exe. Данный модуль установлен на сервере, наблюдаемые телесигналы описаны в конфигурационном файле, который считывается модулем при его запуске из корня программы.

Таким образом, используемые в данный момент способы задания алгоритмов вычисления либо обладают недостаточной функциональностью, имея при этом простой клиентский редактор, либо вообще существуют без графического интерфейса как серверное приложение, выполняющее сложные логические функции.

Для решения этой задачи в данной работе был разработан программный модуль, позволяющий задавать алгоритм вычислений в графическом виде.

Функции программного модуля:

- Создавать алгоритм вычислений графически с помощью функциональных блоков;
- Производить чтение данных НСИ о ТИ и ТС;
- Осуществлять запись алгоритма в XML-файл.

1 Анализ применимости пользовательских интерфейсов

В настоящее время большое количество сфер человеческой деятельности испытывает быстрый рост автоматизации технологических процессов. Вследствие повышения уровня автоматизации, выросло число программного обеспечения, управляющего этими процессами. Для проектирования приложений необходимо большое количество человеческих ресурсов, затраченного времени и существенных финансовых вложений. Разрабатываемые приложения должны быть «дружелюбными», должны соответствовать пожеланиям заказчика и затрачивать наименьшее количество ресурсов компьютера при работе. Сложность создания программного обеспечения заключается в том, что необходимо объединить в себе изложенные выше требования.

Наиболее известным комплексом взаимосвязанных приложений, осуществляющих управление ресурсами компьютера и обеспечивающих взаимодействие с пользователем, является операционная система (ОС). Операционная система скрывает от пользователя подробности взаимодействия с аппаратной частью компьютера, образуя некий слой между ними. Взаимодействие пользователя с операционной системой осуществляется посредством определенных правил, которые должны выполняться как машиной, так и самим пользователем. Используя вышеуказанные правила, можно сформулировать определение интерфейса пользователя.

Пользовательский интерфейс — способ(стандарт) обеспечивающий взаимодействие между пользователем-человеком и программно-аппаратными компонентами компьютерной системы. [2]

1.1 Принципы построения пользовательских интерфейсов

Программное обеспечение разрабатывается с учетом требований и пожеланий пользователя, т.е. должно быть адаптировано под конкретную предметную область. Принципы разработки интерфейса - это

высокоуровневые концепции и представления, которые могут быть приняты к сведению при проектировании программного обеспечения. Необходимо принять один принцип за основополагающий, для разрешения спорных ситуаций. Попытка соблюсти каждый принцип может отрицательно сказаться на конечном результате. Необходимо следовать правилу: «Принципы - это руководство к действию».

Существует три принципа разработки пользовательского интерфейса:

- 1) контроль пользователем интерфейса;
- 2) уменьшение загрузки памяти пользователя;
- 3) последовательность пользовательского интерфейса.

Отметим, что крупные производители операционных систем (Apple, IBM, Microsoft, UNIX), при выпуске своих новых продуктов, публикуют соответствующие руководства и инструкции. В этой документации раскрываются принципы подхода к проектированию интерфейса. Терминология, используемая в разных руководствах может отличаться, но направленность материала, уровень сложности и описанные принципы пользовательского интерфейса имеют единую концепцию. Данные принципы применимы ко всему программному и аппаратному обеспечению, во всех типах и стилях интерфейсов. При составлении принципов необходимо было собрать информацию, чтобы наиболее полно описать каждый из них: производились изыскания в области программного интерфейса, осуществлялись разработки, опрашивались пользователи компьютерных платформ. Данные принципы были приняты и прошли проверку временем с появлением новых компьютерных технологий.

Трактовка этих принципов будет зависеть от аппаратного обеспечения, операционной системы, составляющих пользовательского интерфейса и его задач. Зачастую проектировщики отходят от соблюдения принципов из-за специфики поставленной задачи. Модели пользователя и проектировщика также различны и влияют на то, как будут применяться принципы.

Основные правила проектирования пользовательского интерфейса:

1) дать контроль пользователю;

Опытные проектировщики позволяют пользователям решать некоторые задачи по собственному усмотрению. Это сделано для того, чтобы пользователи имели альтернативу по выполнению функций.

Принципы, которые дают пользователю контроль над системой:

- предоставить пользователю выбор: работать либо мышью, либо клавиатурой, либо их комбинацией;

Следование данному принципу позволяет использовать несколько устройств ввода для реализации одной и той же функции. Например, функция вставки текста в текстовом редакторе Microsoft Office Word.

- позволить пользователю сфокусировать внимание;

Необходимо дать пользователям возможность контролировать процесс работы в программе – каждое событие, которое происходит в системе должно ожидать действия от пользователя.

- демонстрировать сообщения, которые помогут ему в работе;

Во всем интерфейсе использовать понятные для пользователя термины. Следует выбрать манеру построения диалога. Однако не стоит использовать сокращения и излишнее применение разговорных фраз.

- создать условия для немедленных и обратимых действий, а также обратной связи;

Каждое программное обеспечение должно включать в себя функции UNDO и REDO. UNDO – отмена действия, REDO – возврат отмененного действия. Стоит информировать пользователя о том, что данное действие не может быть отменено, и по возможности разрешить ему альтернативное действие.

- приспособливать систему к пользователям с различным уровнем подготовки;

Необходимо предусматривать разделение по опыту пользователей. Не загружать интерфейс различными подсказками опытных пользователей, и не оставлять неопытных пользователей без полезной информации.

- сделать пользовательский интерфейс более понятным;

При похожести интерфейса на большинство программ, которые осуществляют такие же действия, пользователь не будет тратить время на его изучение. Если он разработан неудачно, пользователям придется приложить немало усилий для эффективного использования программного продукта. Задача интерфейса - свободно манипулировать и работать с объектами. Данный интерфейс также называют "прозрачным". "Прозрачность" интерфейса обеспечивается тем, что человеку будет дана возможность пользоваться объектами, отличными от системных команд.

- дать пользователю возможность настраивать интерфейс по своему вкусу;

Пользователь должен быть сосредоточен на выполнении задач, стоящих перед ним, а не изучать функции ПО.

- разрешить пользователю напрямую манипулировать объектами интерфейса;

Простое правило: увеличивать метафоричность, но не ломать ее. Иногда система прямых манипуляций терпит крах, если пользователь не знает, что надо взять и куда это поместить. Единственная проблема прямых манипуляций - визуально не понятно, может ли объект перемещен и отпущен на экране. Пользователи должны чувствовать комфорт при производстве данной операции и знать о возможном результате. Кроме того, необходимо, чтобы интерфейс можно было без труда изучить.

- позволить пользователю думать, что он контролирует ситуацию.

Хорошо разработанный интерфейс должен быть удобен для пользователей и не нагружать его, пока компьютер находится в состоянии загрузки.

2) уменьшить нагрузку на пользователя;

Основанная на знании того, как люди хранят и запоминают информацию, сила компьютерного интерфейса должна защитить память людей от излишней загруженности.

Принципы, позволяющие снизить нагрузку на память пользователя:

- не загружать кратковременную память;

Не вынуждать пользователей запоминать и повторять то, что может (и должен) делать компьютер. Например, когда необходимо заполнить анкету, потребуется ввести некоторые данные, которые фиксируются системой для дальнейшего использования, при повторном входе пользователя в систему или открытия записи. Система должна "запоминать" введенную информацию и обеспечить беспрепятственный доступ к ней в любое время.

- полагаться на распознавание, а не на повторение;

Предусмотреть списки и меню, содержащие объекты или документы, которые можно выбрать, не заставляя пользователей вводить информацию вручную без поддержки системы. Предусмотреть списки наиболее популярных объектов и документов, которые можно просто выбрать без заполнения командных строк и др.

- представить визуальные заставки;

Когда пользователи находятся в каком-то режиме или работают мышью, это должно отразиться на экране. Индикация должна сообщать пользователю о режиме, в котором он находится. Форма курсора может меняться для указания текущего режима или действия, а индикатор - включаться или отключаться. Тест на визуальную информативность продукта: отойти от компьютера во время выполнения задачи и позднее вернуться к работе. Обратит внимание на визуальные подсказки интерфейса, которые должны информировать о том, с чем вы работали, где находились и что делали.

- предусмотреть установки по умолчанию, команды Undo и Rendo;

Использовать способность компьютера сохранять и отыскивать информацию о выборе пользователя, а также о свойствах системы. Предусмотреть многоуровневые системы отмены и повтора команд, обеспечивающие уверенную и спокойную работу с программой.

- обеспечить ярлыки для интерфейса;

Как только пользователи достаточно хорошо осваивают программный продукт, они начинают испытывать потребность в ускорителях. Не игнорировать эту необходимость, однако при разработке следовать стандартам.

- активировать синтаксис действий с объектами;

Объектно-ориентированный синтаксис позволяет человеку понять взаимосвязь между объектами и действиями в программном продукте. Пользователи могут изучать и "листать" интерфейс, выбирая объекты и просматривая доступные действия.

- применять раскрытие и объяснение понятий и действий;

Необходимо позаботиться о легком доступе к часто используемым функциям и действиям. Скрыть редко используемые свойства и функции и позволить пользователю вызывать их по мере необходимости. Использовать вторичные окна.

- увеличить визуальную ясность.

Компьютерные графики и оформители книг хорошо освоили искусство представления информации. Этот навык должны иметь и разработчики пользовательского интерфейса.

3) сделать интерфейс совместимым:

Совместимость - ключевой аспект для использования интерфейса. Однако не следует, во что бы то ни стало стремиться к ней. Одним из основных преимуществ последовательности является то, что пользователи могут перенести свои знания и навыки из старой программы, которой они пользовались раньше, в новую.

Принципы создания совместимости интерфейса:

- проектирование последовательного интерфейса;

Пользователи должны иметь опорные точки при перемещении в интерфейсе. Это заголовки окон, навигационные карты и древовидные структуры. Пользователь также должен иметь возможность завершить поставленную задачу без изменения среды работы или переключения между

стилями ввода информации. Если сначала он использовал клавиатуру, то должна быть обеспечена возможность завершить работу тоже с ней как с главным инструментом для взаимодействия.

- общая совместимость всех программ;

Изучение одной программы не должно кардинально отличаться от изучения подобной программы. Когда похожие объекты не работают одинаково в различных ситуациях, пользователь не понимает происходящего, отрицательно настраивается по отношению к программному продукту.

- сохранение результатов взаимодействия;

Если результаты выполнения операции ПО могут быть отличны от того, что ожидает пользователь, то есть необходимость информировать его перед выполнением действия. Дать ему расширенное описание выполнения действия, возможность отменить действие или произвести другое.

- эстетическая привлекательность и цельность;

Принятый для взгляда интерфейс не должен скрывать недостаток функциональности программного продукта. Пользователи должны получить красивый интерфейс, который поможет им в работе.

- поощрение изучения;

Интерфейсы сегодняшнего и завтрашнего дня - более интуитивны, предсказуемые, дружелюбные, привлекательные. [9]

1.2 Виды пользовательского интерфейса

В настоящее время распространены такие виды пользовательского интерфейса, как:

1. Командный: инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд);
2. Графический: программные функции представляются графическими элементами экрана;
3. Речевой (SLIK-интерфейс): команды подаются голосом путем произнесения специальных зарезервированных слов-команд.

Использование того или иного вида интерфейса в программируемом приложении зависит от задач самого приложения, ресурсах, необходимых для выполнения функций программы, а также подготовкой специалистов, для которых это приложение разрабатывается.

1.2.1 Описание графического интерфейса

Появление интерфейса данного вида совпало с повсеместным использованием операционной системы MS-DOS. Данная ОС способствовала развитию этого интерфейса, а именно улучшение отображения символов и технических параметров монитора.

На первом этапе графический интерфейс был похож на командную строку. Отличия от командного интерфейса заключались в следующем:

1. При отображении символов пользователь мог выделить части символов цветом, дополнить инверсным изображением, подчеркиванием и мерцанием.
2. Изменение стандартного курсора в зависимости от пожелания пользователя.
3. Действие после нажатия любой клавиши зависит от того, в какой части экрана находился курсор.
4. На клавиатуре стали использовать клавиши управления курсором (стрелочки).
5. Уже в этой версии графического интерфейса стали использоваться манипуляторы (мышь, сенсорная панель и т.п.). Они позволяли быстро выделять необходимую область экрана и перемещать курсор.

Подводя итоги, можно привести следующие отличительные особенности этого интерфейса:

1. Выделение областей экрана;
2. Переопределение клавиш клавиатуры в зависимости от контекста;
3. Использование манипуляторов и клавиш клавиатуры для управления курсором;

4. Широкое использование цветных мониторов.

Вторым этапом в развитии графического интерфейса стал интерфейс WIMP (windows, icon, menu, panel). Этот подвид интерфейса характеризуется следующими особенностями:

1. Вся работа с программами, файлами и документами происходит в окнах, ограниченных рамкой;
2. Все объекты представляются в виде значков – иконок. При инициализации значка, кнопки выполняют вполне определенные функции;
3. Все действия с объектами осуществляются с помощью меню;
4. Широкое использование манипуляторов для указания на объекты. Манипулятор становится основным элементом управления.

1.2.2 Описание командного интерфейса

Командный интерфейс использует в качестве единственного способа ввода информации от пользователя к компьютеру клавиатуру, а вывод информации пользователю от компьютера осуществляется с помощью монитора. Эта комбинация «монитор + клавиатура» получила название терминал или консоль.

Команды набираются с помощью командной строки. Командная строка использует символ приглашения и мигающий прямоугольник – курсор. При введении с клавиатуры набора символов на месте курсора появляются символы, а сам курсор сдвигается вправо. Данный принцип похож на набор команды на пишущей машинке. Однако, в отличие от нее, буквы отображаются на дисплее, а не на бумаге, и неправильно набранный символ можно стереть. Команда заканчивается нажатием клавиши ввода. После этого осуществляется переход на начало следующей строки и отображение результата работы команды на мониторе. Затем, если это необходимо, процесс повторяется.

Технология командной строки существует уже достаточно долгое время. Поскольку вводить позволяет только буквы, цифры и знаки препинания, то технические характеристики дисплея являются несущественными. В качестве устройства вывода можно использовать телевизионный приемник или трубку осциллографа.

Более удобным форматом файлов, получившим широкое распространение при работе с командным интерфейсом, стал текстовый формат.

Удачно спроектированные командные интерфейсы обеспечивают:

1. необходимую скорость обработки;
2. эффективную работу с пользователем;
3. экономию ресурсов системы.

1.2.3 Сравнительный анализ командного и графического интерфейсов

Важными преимуществами грамотно разработанных командных языков является их алгоритмическая полнота: данные языки могут применяться для решения любых пользовательских задач, которые не были запланированы разработчиком. Еще одним достоинством командного интерфейса является его жесткий синтаксис, поэтому вероятность произвести ошибочные действия снижена.

Однако существует и ряд недостатков командного интерфейса. Существенным недостатком является необходимость в знании синтаксиса всех команд, а также использовать нужные ключи и опции для них, что не совсем удобно для конечного пользователя, который не сталкивался с технологией командной строки ранее. Также стоит отметить, что текстовое отображение результата выполнения пользовательской задачи является неудобным для анализа полученных значений, а, иногда, практически невозможной, в случае с графикой.

При работе с графическим интерфейсом пользователь взаимодействует с оконным приложением, что удобно при переключении между другими окнами, не теряя полученных в процессе работы данных.

В связи с тем, что графический интерфейс ресурсозависимый, то для своей реализации требует цветной растровый дисплей с высоким разрешением и манипулятор, что не требуется для командных интерфейсов. Программы, ориентированные на графический интерфейс, предъявляют повышенные требования к производительности компьютера, пропускной способности шины, объему его памяти и т.п. Однако этот вид интерфейса интуитивно понятнее и наиболее прост в усвоении. Пользователю достаточно иметь базовый уровень владения компьютером и некоторыми основными приложениями ОС, чтобы обучиться новому приложению. Примером программ с графическим интерфейсом является операционная система Microsoft Windows.

Командные языки требуют затрат времени и усилий для изучения. Графический интерфейс предоставляет новому пользователю возможность быстро оценить доступные возможности и выбрать необходимую. Наглядность интерфейса намного важнее разнообразия возможностей, ведь оно может просто запутать пользователя. Нужно помнить, что пользователь способен одновременно взаимодействовать лишь с довольно ограниченным количеством объектов и параметров.

Относительно высокие накладные расходы, неразрывно связанные с графическими интерфейсами, компенсируются быстротой изучения программного продукта работником и понятностью интерфейса. Нынешние цены на аппаратуру довольно низки, что способствует быстрой окупаемости внедренного графического интерфейса. На фоне системных требований, предъявляемых оперативно-измерительными комплексами, накладные расходы, связанные с проектированием графического интерфейса, оказываются несущественными.

Хорошо продуманный графический интерфейс с грамотно подобранными цветами, красиво отображаемыми управляющими объектами интерфейса программы, сам по себе будет располагать пользователя к взаимодействию и его изучению.

1.3 Анализ графических языков программирования

1.3.1 Виды графических языков программирования

Технологический язык программирования в стандарте МЭК-61131 представляет собой процедурный язык с элементами проблемной ориентированности, специализированной под исполнение в определенной сфере деятельности. Целью технологического языка программирования является предоставление возможности непрофессиональному программисту создавать алгоритмы в своей сфере деятельности.

Первым технологическим языком в сфере автоматике был язык Ladder Logic ориентированный на инженеров-электриков. С его помощью программировались условные и безусловные переходы, используя традиционные для релейных схем обозначения. Собственно, релейные схемы и послужили прообразом этого языка.

В настоящее время стандартом МЭК-61131 описан синтаксис и семантика 5 технологических языков программирования. Среди технологических языков программирования стандарта МЭК-61131[5] выделяют:

1. Ladder Diagram (LD) – это графический язык программирования. Он представляет собой стандартный вариант класса языков релейно-контактных схем. Логические выражения в языке используются в виде реле, которые широко использовались в сфере автоматизации в 60-х годах;
2. Instruction List (IL) – текстовый язык программирования низкого уровня похожий на Assembler;

3. Structured Text (ST) – это текстовый язык высокого уровня общего назначения. По синтаксису схож с Pascal. Самостоятельного значения этот язык почти не имеет, поскольку функционал сильно сокращен. Используется ST совместно с SFC;
4. Function Block Diagram (FBD) – графический язык, по своей сути схожий с LD, однако, вместо реле использует функциональные блоки, схожие с микросхемами;
5. Sequential Function Chart (SFC) – графический язык, который используют для описания алгоритма в виде набора связанных параметров шага (step) и перехода (transition). Шаг – набор операций над переменными, переход – набор условий, определяющих передачу управления следующей паре шаг-переход.

Из описанного выше списка языков визуального программирования лишь 3 являются графическими – LD, FBD, SFC.

Язык LD является графическим языком программирования схем, которые представляют собой аналог релейной схемы. Согласно стандарта МЭК-61131-3, такая форма представления алгоритма облегчит переход инженеров из области релейной автоматики на ПЛК.

К недостаткам данного языка можно отнести то, что LD, построенный по аналогии с релейными схемами, может быть эффективно использован только для описания процессов, имеющих дискретный (двоичный) характер; для обработки «непрерывных» процессов (с множеством аналоговых переменных) такой подход теряет смысл.

Язык FBD является языком графического программирования, аналогично LD, использующий аналогию с электрической схемой. Написанный алгоритм на языке FBD представляет собой совокупность функциональных блоков, входа и выхода которых соединены линиями связи, соединяющие выхода одних блоков с входами других, являются по сути дела переменными программы и служат для пересылки данных между блоками. Каждый блок представляет характеризует математическую операцию

(сложение, умножение, логическую операцию и т.д.) и может иметь, в общем случае, произвольное количество входов и выходов.

Начальные значения переменных задаются с помощью специальных блоков – входов или констант, выходные цепи могут быть связаны, также с блоками вывода.

FBD является наиболее популярным языком стандарта МЭК-61131-3. Графическая форма представления алгоритма, простота в использовании, повторное использование функциональных диаграмм и библиотеки функциональных блоков делают язык FBD незаменимым при разработке достаточно сложных алгоритмов.

Однако, данный язык имеет и недостатки. При описании достаточно большого алгоритма происходит загромождение рабочей области, что способствует большой нагрузке при отыскании нужного участка схемы.

Язык последовательных функциональных схем SFC, использующийся совместно с ST и IL, является языком графического программирования, в котором алгоритм интерпретируется в виде схематической последовательности шагов, объединенных переходами. Язык SFC построен по принципу, близкому к концепции конечного автомата, что делает его одним из самых мощных языков программирования стандарта МЭК 61131-3.

На языке SFC довольно просто и удобно описываются технологические процессы, состоящие из последовательно выполняемых шагов, с возможностью описания нескольких параллельно выполняющихся процессов, для чего в языке имеются специальные символы разветвления и слияния потоков.

Данный язык не лишен и недостатков. Хотя SFC может быть использован для моделирования конечных автоматов, его программная модель не совсем удобна для этого. Это связано с тем, что текущее состояние программы определяется не переменной состояния, а набором флагов активности каждого шага, в связи с чем при недостаточном контроле со

стороны программиста могут оказаться одновременно активными несколько шагов, не находящихся в параллельных потоках.

Еще одно неудобство языка связано с тем, что шаги графически располагаются сверху вниз, и переход, идущий в обратном направлении, изображается в неявной форме, в виде стрелки с номером состояния, в которое осуществляется переход [6].

Проанализировав данные языки, можно сделать вывод, что наиболее подходящий под данную задачу язык – FBD. Он использует в своей концепции функциональные блоки, которые содержат определенные функции: арифметические, логические, тригонометрические и т.д. Прост для понимания, не требует дополнительного времени для изучения.

1.3.2 Пример реализации графических языков программирования

На сегодняшний день существует достаточно много программных комплексов, реализующих функции графических редакторов различными языками графического программирования. Для того, чтобы сформировать требования к разрабатываемому графическому редактору, необходимо проанализировать известные программные комплексы, которые являются наиболее удобными, выполняют схожие функции, используют один из вышеуказанных языков графического программирования и т.д. Для оценки были выбраны следующие комплексы, подходящих под желаемое описание: Simulink, LabView, PLCDesigner.

Simulink – это графическая среда моделирования, позволяющая при помощи блоков строить динамические модели, включая дискретные, непрерывные и гибридные, нелинейные и разрывные системы [8].

Интерактивная среда Simulink, позволяет использовать готовые библиотеки блоков для моделирования электросиловых, механических и гидравлических систем, а также применять развитый модельно-ориентированный подход при разработке систем управления, средств цифровой связи и устройств реального времени.

Дополнительные функции Simulink позволяют решать весь спектр задач от разработки концепции модели до тестирования, проверки, генерации кода и аппаратной реализации. Simulink интегрирован в среду MATLAB, что позволяет использовать встроенные математические алгоритмы, мощные средства обработки данных и научную графику [8].

Simulink использует графический интерфейс и собственный язык визуального программирования, который по своей специфике похож на язык FBD – создание библиотеки функциональных блоков, и использование данной библиотеки при создании алгоритма расчета.

Программа LabVIEW это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G», которая состоит из двух частей:

- блочной диаграммы, описывающей логику работы виртуального прибора;
- лицевой панели, описывающей внешний интерфейс виртуального прибора.

Блочная диаграмма содержит функциональные узлы, являющиеся источниками, приемниками и средствами обработки данных. Также компонентами блочной диаграммы являются терминалы («задние контакты» объектов лицевой панели) и управляющие структуры (являющиеся аналогами таких элементов текстовых языков программирования, как условный оператор «IF», операторы цикла «FOR» и «WHILE» и т. п.). Функциональные узлы и терминалы объединены в единую схему линиями связей [9].

Интегрированная система программирования контроллеров PLCDesigner использует языки технологического программирования в соответствии со стандартом МЭК-61131-3 и обеспечивает разработчика эффективными инструментальными средствами, включающими эмуляцию ПЛК, точки останова, пошаговую отладку, трассировку значений переменных.

Система Designer состоит из интегрированной среды разработки и виртуальной машины, исполняющей программы на целевом контроллере.

Интегрированная среда разработки программ включает в себя редакторы графических и текстовых языков технологического программирования, отладчик, компилятор и загрузчик.

Возможности:

- интуитивно понятный графический интерфейс позволяет легко создавать и отлаживать проекты;
- методология структурного программирования дает возможность описывать автоматизируемый процесс в наглядной и понятной форме;
- механизм объектного визуального программирования позволяет строить алгоритмы технологических процессов в виде логических схем, состоящих из функциональных блоков и связей между ними;
- Наличие встроенных библиотек функций и функциональных блоков существенно сокращает время проектирования.

Особенностью данного приложения является наличие удобных средств разработки и отладки программ:

- редактор дерева проекта;
- редактор переменных;
- графический редактор SFC схем;
- графический редактор FBD схем;
- графический редактор LD схем;
- редактор для программирования на языке ПЛ;
- эмулятор контроллера;
- средства удаленной отладки.

Взяв во внимание, описанное выше, можно сделать вывод, что использование графических редакторов для разработки алгоритмов является основой для многих мощных вычислительных комплексов. Они обладают похожей концепцией – библиотека функциональных блоков, которая используется для создания алгоритмов; соединительные линии связи для придания алгоритму последовательность операций. Ближе всего по описанию и выполняемым функциям, для реализации данного программного модуля

подходит Simulink, т.к. использует графический интерфейс и собственный язык визуального программирования, который по своей специфике похож на язык FBD – создание библиотеки функциональных блоков, и использование данной библиотеки при создании алгоритма расчета.

1.4 Общие требования к графическому редактору

На основе выбранных выше языков графического программирования и интерфейса программы необходимо сформировать общие требования, предъявляемые к разрабатываемому графическому редактору. В качестве документов, используемых при формировании требования, были выбраны основные стандарты IEC-61131-3, IEC-61499, IEC-9241-12.

Общие требования к графическому редактору:

1. Интерфейс должен соответствовать стандартному интерфейсу приложений Windows;
2. Графический интерфейс должен содержать основные объекты данного интерфейса: меню, графические элементы управления, экранные формы, управляющие клавиши и т.д.
3. Функциональные блоки должны иметь возможность взаимодействия с другими функциональными блоками;
4. Библиотека функциональных блоков должна содержать основные функции, необходимые для реализации поставленной задачи;
5. Интерфейс должен быть интуитивно понятен пользователю.

1.5 Вывод по 1 главе

Анализ видов интерфейсов показал, что наиболее подходящим для задач формирования алгоритмов расчетов графически является графический интерфейс, ввиду возможностей пользовательских рабочих мест поддерживать данный интерфейс, удобства представления информации и предоставления пользователю привычное отображение приложения в операционной системе Windows.

Принято решение взять за основу графический язык визуального программирования FBD, который соответствует концепции разрабатываемого программного модуля.

Обзор литературы и поверхностный анализ задачи позволил сформулировать основные программные требования к разрабатываемому графическому редактору.

Для формирования функциональных требований к графическому редактору необходимо определиться с целевой аудиторией, необходимыми функциями, реализующимися в разрабатываемом программном модуле и перечнем взаимодействующих программных модулей.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
5AM5Д	Симонову Игорю Викторовичу

Институт	ЭНИН	Кафедра	ЭСиЭ
Уровень образования	магистратура	Специальность	13.04.02 – Электроэнергетика и электротехника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

<i>1. Стоимость ресурсов научного исследования: материально-технических, энергетических, финансовых, и человеческих</i>	<i>Определение трудовых и денежных затрат, направленных на реализацию исследования. Приблизительная стоимость ресурсов научного исследования, в том числе научно-технического оборудования, составляет около 1,5 млн. руб.</i>
<i>2. Нормы и нормативы расходования ресурсов</i>	<i>Заработная плата участников НИР определяется по тарифной ставке ТПУ</i>
<i>3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования.</i>	<i>Отчисления на социальные нужды 30% от фонда оплаты труда</i>

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Выполнение предпроектного анализа.
2. Инициация проекта
3. Планирование управления научно-техническим проектом
4. Расчет сметы затрат на исследование

Перечень графического материала (с точным указанием обязательных чертежей)

1. Иерархическая структура работ.
2. Календарный план-график научно-исследовательского проекта.

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент кафедры менеджмента	Фигурко А. А.	К.Э.Н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
5AM5Д	Симонов Игорь Викторович		

4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Научно-исследовательская работа (НИР) включает в себя создание программного модуля «Графический интерфейс для программы дорасчета телеизмерений (ТИ) и телесигнализации (ТС)». В работе рассматриваются требования к функциональным возможностям программного обеспечения, возможность реализации функциональных блоков расширенной функциональности.

Целью данного раздела является определение потенциала НИР, планирование процесса управления исследованием, определение ресурсной, финансовой и экономической эффективности.

Достижение цели обеспечивается решением задач:

- разработка общей экономической идеи проекта, формирование концепции проекта;
- организация работ по научно-исследовательскому проекту;
- планирование научно-исследовательских работ;
- определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.

С учетом решения данных задач была сформирована структура и содержание данного раздела.

4.1 Предпроектный анализ

4.1.1 Анализ конкурентных технических решений и потенциальные потребители результатов исследования

АО «СО ЕЭС» (Акционерное общество «Системный оператор Единой энергетической системы») России осуществляет централизованное оперативно-диспетчерское управление, обеспечивает перспективное развитие ЕЭС России. Для достижения необходимой надежности оперативного управления режимами энергосистем, диспетчерам должны быть доступны все

необходимые данные, предоставляемые оперативно-информационным комплексом (ОИК) СК-2007 в виде таблиц и графиков на формах отображения. Однако не всегда данные, поступающие с устройств телемеханики, являются достаточными, и приходится производить дорасчет недостающих параметров. Задание алгоритмов, по которым производится расчет требуемых величин, осуществляется в Редакторе формул ОИКа посредством письменного написания формул. Данный редактор обладает большим количеством функций (арифметическими, логическими, тригонометрическими и т.д.) Однако для реализации более сложного функционала, например, счетчик с заданным временем отсчета, данный редактор является недостаточным.

Данный программный модуль создается с целью задания алгоритмов работы с помощью функциональных блоков, реализации сложных алгоритмов для специалистов службы автоматизированных систем диспетчерского управления. Использование графического интерфейса позволит сократить трудозатраты специалистов как при проектировании новых алгоритмов, так и редактировании уже имеющихся.

4.2 Инициация проекта

Группа процессов инициации состоит из процессов, которые выполняются для определения нового проекта или новой фазы существующего. В рамках процессов инициации определяются изначальные цели и содержание проекта [11]. Цели и результаты проекта представлены в таблице 1.

Таблица 1 – Цели и результат проекта

Цели проекта:	Создание графического интерфейса для программы дорасчета телеизмерений (ТИ) и телесигнализации (ТС)
Ожидаемые результаты проекта:	Возможность задания алгоритмов работы в графическом виде с помощью функциональных блоков
Требования приемки результата проекта:	<ol style="list-style-type: none"> 1. Графическая отрисовка функциональных блоков и связей между ними. 2. Выполнение основных функций редактора: копирование, удаление и т.д. 3. Сохранение алгоритма расчета в XML-файл. 4. Отправка данного файла в программу дорасчета ТИ и ТС. 5. Чтение XML-файла с алгоритмом расчетов для правки

4.3 Планирование управления научно-техническим проектом

Группа процессов планирования состоит из процессов, осуществляемых для определения общего содержания работ, уточнения целей и разработки последовательности действий, требуемых для достижения данных целей.

4.3.1 Иерархическая структура работ проекта

Иерархическая структура работ (ИСР) – детализация укрупненной структуры работ. В процессе создания ИСР структурируется и определяется содержание всего проекта.

В зависимости от последовательности их применения для одного проекта можно построить несколько вариантов ИСР:

1. продуктовый подход;
2. подход по жизненному циклу;
3. функциональный подход;
4. организационный подход;
5. смешанный подход.

При составлении структуры ИСР используем смешанный подход, первый уровень – подход по жизненному циклу, второй уровень – функциональный подход. Структура ИСР представлена на рисунке 12.

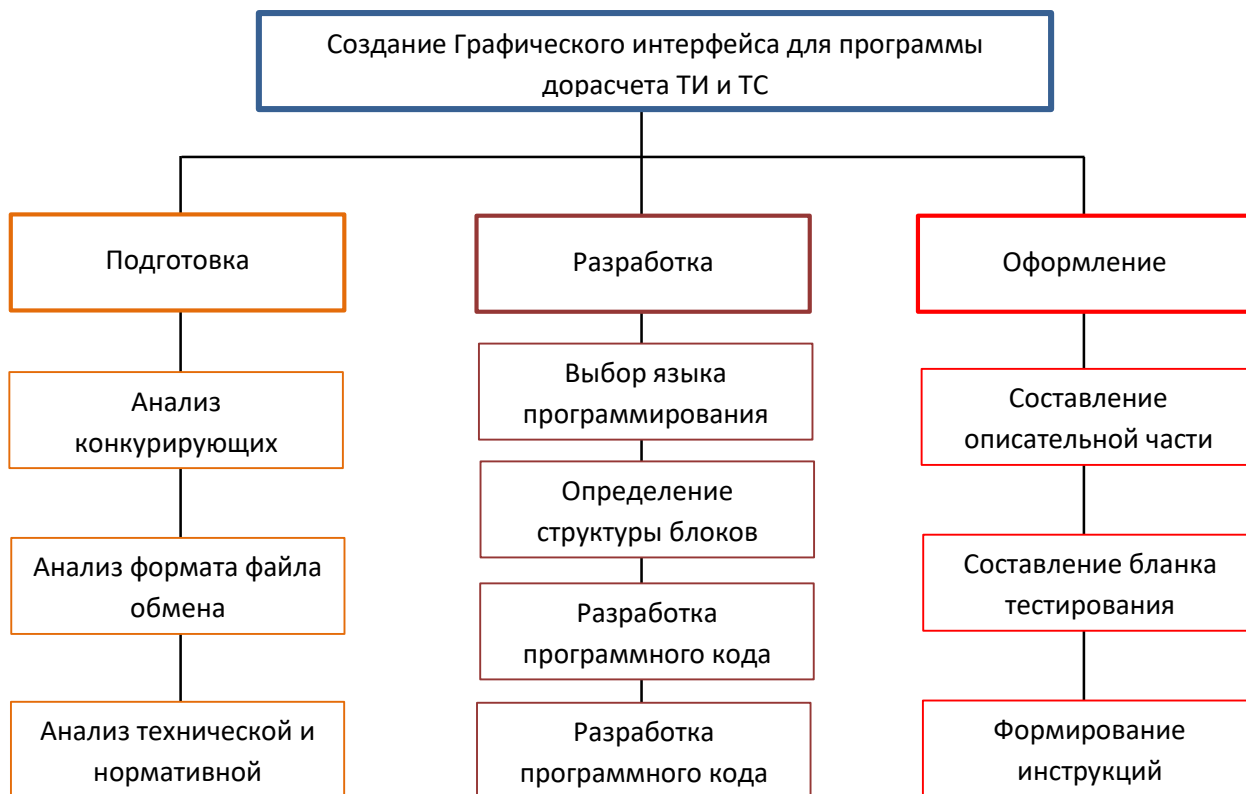


Рисунок 12 – Иерархическая структура работ

4.3.2 Организация и планирование работ

При выполнении проектов необходимо оптимально планировать сроки проведения каждого из этапов работ и занятость каждого из участников. Для этого составляется полный перечень проводимых работ и определяются продолжительность каждого этапа и исполнители. Результатом планирования работ является сетевой, либо календарный (линейный) график реализации проекта. Наиболее наглядным и удобным способом для организации проекта является использование календарного графика. Для его построения составляется перечень работ и соответствие работ своим исполнителям, определяются даты начала и окончания работ. Основные исполнители в проекте: научный руководитель (Р) и инженер (И). Линейный график представлен в виде таблицы 2.

Таблица 2 – Линейный план проекта

Название задачи	Длительность	Начало	Окончание	Состав участников
Планирование работ	137 дней	01.10.15	15.02.16	
Выбор актуальной темы	115 дней	01.10.15	24.01.16	Р
Составление задания	7 дней	25.01.16	01.02.16	Р
Получение задания	1 день	01.01.16	01.01.16	И
Составление плана работ	14 дней	01.02.16	15.02.16	Р, И
Подготовка к разработке	105 дней	16.02.16	31.05.16	
Поиск и изучение нормативно-технической литературы	14 дней	16.02.16	01.03.16	Р, И
Анализ требований к графическому интерфейсу	21 день	01.03.16	23.03.16	Р, И
Определение языка и платформы программирования	3 дня	23.03.16	27.03.16	И
Изучение особенностей графического программирования	42 дня	27.03.16	09.05.16	И
Формирование перечня параметров, необходимых для реализации функциональных блоков	7 дней	09.05.16	17.05.16	И
Изучение взаимодействия серверного и клиентского программного обеспечения	13 дней	18.05.16	31.05.16	И
Реализация (разработка) ПО	187 дней	01.09.16	10.06.17	
Кодирование функциональных блоков: создание, отрисовка, перетаскивание	98 дней	01.09.16	08.12.16	И
Особенности формирования XML-файла с алгоритмом расчетов	14 дней	09.12.16	23.12.16	И
Чтение данных НСИ с БД	16 дней	30.01.17	15.02.17	И
Взаимодействие с серверным приложением дорасчета ТИ и ТС	78 дней	13.03.17	30.05.17	И
Расширение библиотеки функциональных блоков	31 день	10.05.17	10.06.17	И
Тестирование и отладка	31 день	10.05.17	10.06.17	И
Составление раздела социальной ответственности и финансового менеджмента	14 дней	28.04.17	12.05.17	
Составление раздела социальная ответственность	7 дней	28.04.17	05.05.17	И
Составление раздела финансового менеджмента	7 дней	05.05.17	12.05.17	И
Оформление расчетно-пояснительной записки	183 дней	03.10.16	01.06.17	

На основе календарного плана проекта построена Диаграмма Гантта (приложение А). Диаграмма Гантта – это тип столбчатых диаграмм, который используется для иллюстрации календарного плана проекта, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

4.4 Расчет сметы затрат на исследование

В состав затрат на создание проекта включается стоимость всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости на выполнение данной разработки производится по следующим статьям затрат:

- материалы и покупные изделия;
- основная заработная плата;
- отчисления в социальные фонды;
- расходы на электроэнергию;
- амортизационные отчисления;
- прочие расходы.

К статье расходов «Затраты на материалы» относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ [11]. Цена материальных ресурсов определяется по средней рыночной стоимости на 2015 год по соответствующим ценникам [12, 13] и приведена в таблице 3.

Таблица 3 – Затраты на материалы и оборудования

Наименование материалов	Цена за ед., руб.	Кол-во, шт.	Сумма, руб.
Офисная техника			
Компьютер	27000	1	27000
Мышка	400	1	400
Клавиатура	300	1	300
Офисная мебель			
Стол	3000	1	3000

Стул	1500	1	1500
Программное обеспечение			
Microsoft Office 2013	5000	1	5000
Microsoft Visual Studio	10000	1	10000
Итого	47 200		

Расходы на материалы составили $C_{\text{мат}} = 47\,200$ руб.

Расчет основной заработной платы включает заработную плату научного руководителя и инженера, а также премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Среднедневная заработная плата рассчитывается по формуле:

$$\text{Дневная з/плата} = \frac{\text{Месячный оклад}}{20,58 \text{ дней}}$$

Расчеты затрат на основную заработную плату приведены в таблице 4. При расчете учитывалось, что в году 247 рабочих дня и, следовательно, в месяце 20,58 рабочих дня. Затраты времени на выполнение работы по каждому исполнителю определены при составлении календарного графика проекта (таблица 2). Также был принят во внимание коэффициент, учитывающий коэффициент по премиям $K_{\text{ПР}} = 0,1$ и районный коэффициент $K_{\text{РК}} = 0,3$. Итоговый коэффициент $K = 1 + K_{\text{ПР}} + K_{\text{РК}} = 1 + 0,1 + 0,3 = 1,4$.

Таблица 4 – Затраты на основную заработную плату

Исполнитель	Оклад, руб./мес.	Месячная з/пл. работника, руб.	Среднедневная ставка, руб./день	Затраты времени, дни	Коэфф и-циент	Фонд з/платы, руб.
Научный руководитель	28194,00	39471,6	1917,96	171	1,4	459151,624
Инженер	14099,00	19738,6	959,12	429	1,4	576047,472
Итого						1035199,096

Таким образом, затраты на основную заработную плату составили $C_{осн} = 1\,035\,199,096$ руб.

Затраты по статье отчислений от заработной платы составляют отчисления во внебюджетные фонды.

Отчисления по заработной плате определяются по следующей формуле:

$$C_{соц} = K_{соц} \cdot C_{осн}$$

где $K_{соц}$ – коэффициент, учитывающий размер отчислений из заработной платы. Данный коэффициент составляет 30,2% от затрат на заработную плату и включает в себя:

- отчисления в пенсионный фонд;
- на социальное страхование;
- на медицинское страхование.

Отчисления из заработной платы составили:

$$C_{соц} = 0,302 \cdot 1\,035\,199,096 = 312\,630,1 \text{ (руб.)}$$

Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию при работе оборудования, а также затраты на электроэнергию, потраченную на освещение. Затраты на электроэнергию при работе оборудования для технологических целей рассчитываются по формуле:

$$\mathcal{E}_{об} = P_{об} \cdot Ц_{э} \cdot t_{об}$$

где $\mathcal{E}_{об}$ – затраты на электроэнергию, потребляемую оборудованием, руб.;

$P_{об}$ – мощность, потребляемая оборудованием, кВт;

$Ц_{э}$ – тарифная цена за 1 кВт·час, $Ц_{э} = 2,17$ руб/кВт·час;

$t_{об}$ – время работы оборудования, час.

Время работы оборудования вычисляется на основе данных для инженера из календарного плана проекта, учитывая продолжительность рабочего дня равна 8 часов.

$$t_{об} = T \cdot t_{сут},$$

где $t_{сут}$ – длительность работы освещения за смену, час;

T - время, затраченное на проведение работ, $T = 429$ дня.

Таким образом время работы оборудования $t_{ОБ} = 429 \cdot 8 = 3432$ час.

Мощность, потребляемая оборудованием РОБ, принимаем равную мощности блока питания компьютера и монитора:

$$P_{ОБ} = P_{БП} + P_{М},$$

$$P_{ОБ} = 0,4 + 0,1 = 0,5 \text{ кВт},$$

$$\mathcal{E}_{ОБ} = 0,5 \cdot 1,8 \cdot 3432 = 3089 \text{ руб.}$$

Затраты на электроэнергию для технологических целей приведены в таблице 5.

Расчета затрат на электроэнергию для освещения помещения, где осуществлялось выполнение проекта, рассчитывается по формуле:

$$\mathcal{E}_{ОС} = P_{ОС} \cdot \text{Ц}_{\mathcal{E}} \cdot t_{ОС},$$

где $\mathcal{E}_{ОС}$ – затраты на электроэнергию для освещения, руб.;

$t_{ОС}$ – время работы осветительных приборов, час;

$P_{ОС}$ – мощность, потребляемая осветительными приборами, кВт.

Мощность, потребляемая освещением, определяется по формуле:

$$P_{ОС} = P_{УСТ.ОС} \cdot K_C,$$

где $P_{УСТ.ОС}$ – установленная мощность осветительных приборов, принимаем $P_{УСТ.ОС} = 0,216$ кВт;

K_C – коэффициент спроса, зависящий от количества, загрузки групп электроприемников, для внутреннего освещения $K_C = 0,9$.

Таким образом, потребляемая мощность осветительных приборов составляет:

$$P_{ОС} = 0,216 \cdot 0,9 = 0,194 \text{ кВт}.$$

Затраты на электроэнергию для освещения помещения составили:

$$\mathcal{E}_{ОС} = 0,194 \cdot 2,17 \cdot 3432 = 1444 \text{ руб.}$$

Затраты на электроэнергию для освещения помещения представлены в таблице 5.

Общие затраты на электроэнергию определяются по формуле:

$$\mathcal{E} = \mathcal{E}_{ОБ} + \mathcal{E}_{ОС},$$

Результаты расчета затрат на электроэнергию представлены в таблице 5

Таблица 5 – Затраты на электроэнергию

Наименование оборудования	Время работы оборудования t , час	Потребляемая мощность P , кВт	Затраты Э, руб.
Затраты на электроэнергию для технологических целей			
Персональный компьютер	3 432	0,5	3 089
Затраты на электроэнергию для освещения помещения			
Освещение	3 432	0,828	1 444
Итого			4 533

Расчет амортизационных расходов

В статье «Амортизационные отчисления» от используемого оборудования рассчитывается амортизация за время выполнения работы для оборудования, которое имеется в наличии.

Амортизационные отчисления рассчитываются на время использования персонального компьютера по формуле:

$$C_{AM} = \frac{H_A \cdot C_{OB}}{F_D} \cdot t_{BT} \cdot n,$$

где H_A – годовая норма амортизации, $H_A = 25\%$;

C_{OB} – цена оборудования, $C_{OB} = 27\,700$ руб.;

F_D – действительный годовой фонд рабочего времени, $F_D = 2416$ часов;

t_{BT} – время работы вычислительной техники при создании программного продукта, $t_{BT} = 3432$ часа;

n – число задействованных ПЭВМ, $n = 1$.

Затраты на амортизационные отчисления составили:

$$C_{AM} = \frac{0,25 \cdot 27700}{2416} \cdot 3432 \cdot 1 = 9\,839 \text{ руб.}$$

Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на разработку проекта, которые не учтены в предыдущих статьях [11].

Прочие расходы составляют 16% от единовременных затрат на выполнение технического продукта и проводятся по формуле:

$$C_{\text{ПРОЧ}} = (C_{\text{МАТ}} + C_{\text{ОСН}} + C_{\text{СОЦ}} + Э + C_{\text{АМ}}) \cdot 0,16,$$

$$C_{\text{ПРОЧ}} = (47200 + 1035199,096 + 312630 + 4533 + 9839) \cdot 0,16 = 225\,504 \text{ руб.}$$

Расчет общей себестоимости разработки

Проведя расчет сметы затрат на разработку, можно определить общую стоимость разработки проекта «Макет демонстрационной модели принципов КТ».

Таблица 6 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{МАТ}}$	47 200
Фонд оплаты труда	$C_{\text{ОСН}}$	1 035 200
Отчисления из ФОТ	$C_{\text{СОЦ}}$	310 559
Расходы на электроэнергию	Э	4 533
Амортизационные отчисления	$C_{\text{АМ}}$	9 839
Работы, выполняемые подрядчиком	$C_{\text{СТОР}}$	-
Прочие расходы	$C_{\text{ПРОЧ}}$	225 504
Итого		1 408 060

Таким образом, расходы на разработку составили $C = 1\,408\,060$ руб.

4.5 Вывод по 4 главе

В ходе выполнения данного раздела определен потенциал НИР, а именно построена иерархическая структура работ, с помощью которой участники проекта могут наглядно можно определить объем работ, понятен конечный результат работы и способы его достижения.

При выполнении проектов необходимо оптимально планировать сроки проведения каждого из этапов работ и занятость каждого из участников. Для этого составлен полный перечень проводимых работ и определена продолжительность каждого этапа. В результате планирования построен календарный график реализации проекта, с помощью которого возможно наглядно определить временные характеристики работ.

Расходы на разработку программного модуля составили 1 408 060 руб.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Институт	ЭНИН
Направление подготовки (специальность)	13.04.02 – Электроэнергетика и электротехника
Уровень образования	Магистратура
Кафедра	ЭСиЭ
Период выполнения	(осенний / весенний семестр 2016/2017 учебного года)

Студенту:

Группа	ФИО
5АМ5Д	Симонов Игорь Викторович

Тема работы:

Создание графического интерфейса для программы дорасчета телеизмерений и телесигнализации	
Утверждена приказом проректора-директора (директора) (дата, номер)	626/с

Форма представления работы:

Магистерская диссертация (бакалаврская работа, дипломный проект/работа, магистерская диссертация)

ЗАДАНИЕ

Перечень вопросов, подлежащих исследованию, проектированию и разработке	<ol style="list-style-type: none"> 1. Анализ основных причин нарушения работы программного обеспечения. 2. Анализ причин, приводящих к неправильному функционированию ОИК СК-2007. Возможные пути решения причин некорректной работы программного обеспечения. 3. Анализ причин, приводящих к неправильному функционированию программного модуля Графический интерфейс для программы дорасчета телеизмерений и телесигнализации. Мероприятий для обеспечения надежной работы разрабатываемого программного модуля.
Перечень расчётного и графического материала	

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Невский Егор Сергеевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
5АМ5Д	Симонов Игорь Викторович		

5 Социальная ответственность

При создании программного продукта, вне зависимости от его назначения, необходимо свести к минимуму случаи некорректной работы программы. С целью исключения возможных неблагоприятных сценариев работы алгоритмов программы должны быть предусмотрены все возможные варианты и конфигурации исходных данных, запрет на использование данных с непозволительной структурой или некорректными значениями, сведение влияния пользователя на результаты работы программы к минимуму.

Результат некорректной работы программного обеспечения (ПО) «Графический интерфейс для программы дорасчета телеизмерений(ТИ) и телесигналов(ТС)», способного задавать алгоритм расчетов в графическом виде, может иметь негативное влияние на оперативно-информационный комплекс (ОИК) СК-2007.

Задачей данной главы является определение опасностей при сбое работы программного обеспечения; рассмотрение способов и методов, позволяющих минимизировать случаи некорректной работы разработанного программного модуля.

5.1 Этапы разработки программы

Разработка программного обеспечения включает три этапа [3]:

- 1) проектирование (постановка задачи, построение математической модели, разработка алгоритма);
- 2) кодирование (написание программного кода);
- 3) отладка.

Проектирование выполняется с целью получения алгоритма решения задачи по представлению, переработке информации. Целью этапа кодирование является запись полученного алгоритма на выбранном языке (языках) программирования. Отладка включает тестирование ПО и исправление ошибок, найденных в результате тестирования. Только для

некоторых программ этапы разработки программы выполняются строго последовательно во времени (рисунок 13).

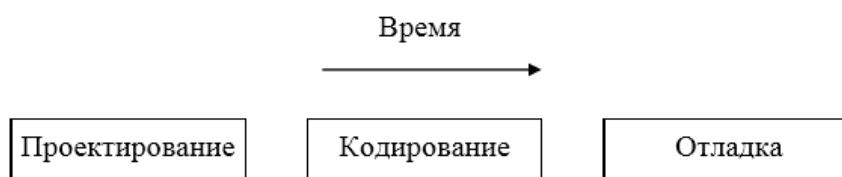


Рисунок 13 – Последовательное выполнение этапов разработки программы

Для большинства ПО эти этапы выполняются параллельно (одновременно). То есть одновременно одна часть программы проектируется, другая ее часть кодируется, а третья – отлаживается. Естественно, что проектирование при этом несколько опережает этап кодирования, а кодирование – отладку (рисунок 14).



Рисунок 14 – Параллельное выполнение этапов разработки программы

Существующие технологии программирования различаются между собой реализацией указанных трех этапов. Главным из них является этап проектирования. Если этапу проектирования было уделено достаточно много времени, что позволило выбрать лучшее сочетание алгоритмов, то этапы кодирования и отладки выполняются намного быстрее. И наоборот, при недостаточно продуманном алгоритме на этапе проектирования ПО, этапы кодирования и отладки значительно усложняются.

После завершения разработки программы начинается ее сопровождение. Во время сопровождения решаются две основные задачи:

- 1) исправление выявляющихся в ходе эксплуатации ошибок;
- 2) корректировка программы, вызванная изменениями в решаемой ею задаче по переработке информации.

Существуют различные типы программных ошибок [4], которые могут возникать на этапе разработки программы программного обеспечения.

Наиболее распространённая и наиболее серьезная из всех ошибок – логическая. Когда написанная программа на любом языке компилирует и работает правильно, но выдает неправильный результат, ошибка заключается в описании логики ПО. Это ошибка наследуется от неточностей в базовом алгоритме. Отсюда можно сделать вывод, что ПО работает неправильно, а значит является неработоспособным. При исправлении такой ошибки, необходимо фундаментальное изменение алгоритма, которое в свою очередь повлечет изменение алгоритмов, сопряженных с данным.

5.2 Причины, приводящие к неправильному функционированию ОИК СК-2007

Оперативно-информационный комплекс СК-2007 представляет собой объединение аппаратной части и программного обеспечения, содержащего в себе большое количество модулей, выполняющих самые разнообразные функции для мониторинга и контроля энергосистемы.

Для оценки качества работы оперативно-информационного комплекса и нахождения ошибок, возникающих в процессе эксплуатации, энергетические компании уже долгие годы проводят многократные испытания каждого компонента СК-2007. В дальнейшем, компания-производитель данного комплекса, с помощью обратной связи с пользователями, производит сбор информации о возникающих проблемах в работе СК-2007. Посредством отправки пользователям обновлений программы происходит устранение причин некорректной работы комплекса и, тем самым, повышается качество выполняемых им функций.

Долговременное использование ОИК СК-2007 в СО ЕЭС показывает, что данный комплекс практически лишен недостатков в своих алгоритмах по выполнению различного рода задач. Кроме того, налажена функция контроля и информирования персонала о некорректном вводе данных. На данный

момент обновления, выпускаемые производителем, касаются исправления неточностей в работе вторичных модулей, которые незначительно влияют на работу основных, а также оптимизации работы всего комплекса. Однако при изменении алгоритма работы и влиянии человеческого фактора разработанные новшества могут положительно сказаться на работе одних компонентов и негативно на других. Например, при увеличении быстродействия комплекса в целом, произойдет значительное сокращение сроков хранимой информации, что недопустимо для энергетических компаний.

Помимо вышеперечисленных причин отказ работы СК-2007 может произойти по двум факторам, от которых тяжело защитить комплекс. К ним относятся:

- Ошибочные действия пользователя;
- Неисправность аппаратных средств информационной системы.

Отсутствие внимания или достаточного опыта у эксплуатирующего персонала может привести к нарушению функционирования комплекса, а вся используемая аппаратура имеет свой рабочий ресурс. Для устранения подобных причин в ОИК СК-2007 предусмотрено резервирование, благодаря которому, неисправность какой-либо из частей комплекса не влечет к полному останову технологического процесса, а система авторизации повышает ответственность каждого пользователя при работе с комплексом.

5.3 Причины, приводящие к неправильному функционированию графического интерфейса

Программный модуль Графический интерфейс для программы дорасчета ТИ и ТС предназначен для задания в графическом виде алгоритмов обработки ТИ и ТС и передачи данного алгоритма серверному модулю Дорасчет ТИ и ТС.

Неисправная работа или останов Графического интерфейса не повлияет на работу СК-2007, так как является клиентским приложением, и все манипуляции с ТИ и ТС пользователь ведет на рабочем месте.

Помимо разработки самого алгоритма работы необходимо учитывать «слабые места» программы. Их анализ помогает на начальном этапе проектирования предотвратить возможные ошибки системы, приводящие к сбоям в работе ПО. К возможным причинам нарушения функционирования Графического интерфейса для программы дорасчета ТИ и ТС относятся:

- Неправильные действия персонала. Действия пользователя, противоречащие логике программы не должны исполняться, однако, программный модуль должен отреагировать на это действие сообщением об невозможности исполнения данного действия. Устранение этой причины заключается в разработке удобного пользовательского интерфейса с обратной связью, который интуитивно понятен, отвечает всем современным требованиям эргономики;

- Проблемы с вводом-выводом информации. При некорректном внесении информации происходит оповещение о неправильно вводе данных. Вывод информации из Графического интерфейса для программы дорасчета ТИ и ТС осуществляется в виде XML-файла, в котором содержатся алгоритмы расчета. При неправильной записи алгоритма в XML-файл, серверное приложение не сможет правильно обработать данный файл, что может повлечь за собой останов серверного приложения. Для исключения возникновения данной ошибки предусмотрена проверка XML-файла на целостность, а также конфигурирование данного файла в строгой последовательности;

- Неисправность аппаратных средств, которая привела бы к нарушению функциональности СК-2007 крайне мала, так как для работы программного модуля требуется персональный компьютер (рабочее место пользователя);

- Завершение программы при потере электроснабжения рабочего места пользователя. Внештатное завершение работы Графического редактора никак не повлияет на функционирование серверного приложения и СК-2007.

Однако, если данное завершение работы модуля произойдет в момент отправки XML-файла с алгоритмом работы, то это вызовет останов серверного приложения. Во избежание такого рода ошибок предусмотрено последовательное исполнение команд: сначала конфигурация XML-файла, а затем его отправка с помощью нажатия на другую исполняющую кнопку;

- Ошибки, скрытые в самой программе, являются наиболее значительной причиной неправильной работы программы. Устранение этой проблемы ведется с помощью тщательной проработки функционала программы, определения всего объема и типа данных, затрагиваемых в ПО, а также детальной проработки алгоритмов работы программы.

Для реализации мероприятий, касающихся надежной работы программного модуля, технология разработки программного обеспечения должна удовлетворять следующим требованиям [1]:

1. Необходима стандартизация языков проектирования программ, оформления и испытания программных модулей, а также гарантии их качества.

Программа разработана на языке C#. В настоящее время C# реализует современные принципы проектирования, сочетающиеся с современными принципами объектно-ориентированного программирования, поэтому имеет хорошие перспективы использования в науке и промышленности.

2. Ведение постоянного контроля и обеспечение качества программы.

По результатам тестирования и опытной эксплуатации программы необходимо выявлять случаи некорректной работы программы и устранить возникшие дефекты. Так же для обеспечения удобства использования функционала программы доработать интерфейс с учетом замечаний пользователей.

3. Пользователю программы необходимо дать четкое представление о возможностях данной программы и технологических условиях эксплуатации, при которых гарантируются определенные функции и качества.

Пользователь обеспечен информационно-справочной системой в объеме, необходимом для освоения инструментария программы и создания условий для корректной обработки исходных данных.

4. Необходимо вести фиксацию и документирование всех этапов разработки для облегчения последующей доработки программного модуля.

При разработке программного обеспечения использовались средства фиксации в хронологическом порядке всех действий, выполняемых в процессе изготовления программного изделия. Эти средства позволяют восстанавливать любое состояние процесса разработки на любом интервале изготовления ПО.

5.4 Вывод по 5 главе

Анализ основных ошибок работы ПО, описание которого содержит данная глава, способствовал выявлению уязвимостей в программе, которые могут повлиять на корректность работы разрабатываемого программного модуля. Основной уязвимостью программы является наличие ошибок и неточностей в коде программы. Обнаружение и исправление данного вида ошибок достигается многократными проверками и тестированиями.

Основной проблемой при возникновении ошибок в программе является влияние данного программного модуля на работу серверного приложения Дорасчет ТИ и ТС. Для минимизации негативного влияния на серверное приложение при возникновении ошибок Графического редактора, необходимо комплексное тестирование данных программных модулей для выявления ошибок, унификация взаимодействующих между ними файлов.

Во избежание негативных последствий при некорректной работе алгоритмов программы были реализованы программные мероприятия, сводящие к минимуму эффект влияния человеческого фактора и возникновение непредвиденных сценариев работы алгоритмов ПО.

Заключение

Цель данной работы заключалась в разработке программного модуля, позволяющего создавать алгоритмы дорасчета в графическом редакторе при помощи функциональных блоков.

Для удобства представления информации и проектирования алгоритмов расчета наиболее целесообразным является язык графического программирования FBD, который основывается на создании функциональных блоков. Каждый блок имеет свое обозначение, набор параметров и выполняет заложенные функции.

Программный модуль разработан в соответствии с основными принципами построения графического интерфейса.

Разработанный программный модуль позволяет создавать функциональные блоки, используя библиотеку, считывать нормативно-справочную информацию в части списка телеизмерений и телесигнализации, производить запись и чтение алгоритма расчета, используя файл формата XML.

Разработанный программный модуль полностью удовлетворяет поставленным требованиям в части реализации основных функций редактора, соответствия принципам построения пользовательских интерфейсов.

Разработка и внедрения в эксплуатацию программного модуля «графический интерфейс» является экономически обоснованной, о чем свидетельствует выполненный предпроектный анализ, инициация проекта, планирование управления научно-техническим проектом, расчет сметы затрат на исследование.

Рассмотрены вопросы социальной ответственности, к которым относится анализ основных причин нарушения работы программного модуля, рассмотрение угроз при возникновении сбоев работы программы, избежание негативных последствий при некорректной работе алгоритмов программного модуля.

Список используемых источников

- 1 Оперативно-информационный комплекс СК-2007С. Том I. Архитектура. ред. 26 от 08.11.2013г – ЗАО «Монитор Электрик», 2013.
- 2 ISO/IEC/IEEE 24765:2010 Systems and software engineering — Vocabulary
- 3 Электронный ресурс: функции СК-2007, режим доступа <http://www.monitel.ru/products/ck-2007c/overview/funksii-scada.html> (дата обращения 22.11.2016)
- 4 Мандел Т. Разработка пользовательского интерфейса: пер с англ. – М: ДМК Пресс, 2001. – 416 с., ил. (Серия «Для программистов»).
- 5 Электронный ресурс: статья о выборе языка программирования , режим доступа <https://www.ibm.com/developerworks/ru/library/wa-optimal/> (дата обращения 07.05.2016)
- 6 ГОСТ Р МЭК 61131-3-2016 Контроллеры программируемые. Часть 3. Языки программирования
- 7 IEC 61499-1 Function blocks – Part 1: Architecture - Edition 2.0
- 8 С.В. Одиночкина. Основы технологий XML - СПб: НИУ ИТМО, 2013. – 56 с.
- 9 Смирнова М.В. Основные правила и принципы построения графического пользовательского интерфейса: сведения, относящиеся к заглавию / М.В. Смирнова // Актуальные проблемы гуманитарных и естественных наук. – 2016. – 2-2. – 88-93
- 10 Электронный ресурс: описание программного обеспечения, режим доступа <http://matlab.ru/products/simulink> (дата обращения 15.05.2017)
- 11 Электронный ресурс: описание основных функций ПО, режим доступа <http://russia.ni.com/labview> (дата обращения 15.05.2017)
- 12 Гаврикова Н.А. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / Н.А. Гаврикова, Л.Р. Тухватулина, И.Г. Видяев, Г.Н. Серикова, Н.В. Шаповалова;

- Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 73 с.
- 13 Электронный ресурс: прайс-лист фирмы DNS, режим доступа <http://www.dns-shop.ru> (дата обращения: 02.05.2017).
 - 14 Электронный ресурс: прайс-лист фирмы SoftLine, режим доступа <http://store.softline.ru/microsoft/microsoft-visual-studio-professional/> (дата обращения: 02.05.2017).
 - 15 Электронный ресурс: прайс-лист фирмы ИКЕА, режим доступа http://gibtech.ru/blog/discus?entry_id=177 (дата обращения: 02.05.2017).
 - 16 Размер окладов ППС и НС ТПУ.
 - 17 Королёва Н.И Стратегический менеджмент: Учебное пособие. Томск Издание ТПУ, 2005г.
 - 18 ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. – М.: ИПК Издательство стандартов, 1994. – 12 с.
 - 19 Плаксин М.А. Тестирование и отладка программ для профессионалов будущих и настоящих. – М.: Бином, 2013. – 168 с.
 - 20 Черкесов Г.Н. Надежность аппаратно-программных комплексов. – СПб.: Питер, 2005. – 478 с.

Приложение А (обязательное)

Календарный план-график научно-исследовательского проекта

