МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра «Электронные вычислительные машины»

|  |  |
| --- | --- |
|  |  |

Методические указания к выполнению отчетов о преддипломной практике

Отчет

о преддипломной практике

|  |  |
| --- | --- |
|  | Руководитель практики,  к.т.н., доцент каф. ЭВМ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.О. Фамилия  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г.  Выполнил,  студент группы КЭ-222  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.О. Фамилия  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г. |

Челябинск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное   
образовательное учреждение высшего образования   
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра «Электронные вычислительные машины»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заведующий кафедрой ЭВМ  \_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Топольский  «\_\_\_»\_\_\_\_\_\_\_\_\_\_2022 г. |

**ЗАДАНИЕ**

**на преддипломную практику магистра**

студенту группы КЭ-222

Фамилия Имя Отчество

обучающемуся по направлению

09.04.01 «Информатика и вычислительная техника»

1. **Тема работы: «**Методические указания к выполнению отчетов о преддипломной практике» утверждена приказом по университету от 25 декабря 2019 г. №1234
2. **Срок сдачи студентом законченной работы:** 1 марта 2022 г.
3. **Исходные данные к работе:** статьи, книги, техническое задание.

*Пример 1.*

* Заенцев, И.В. Нейронные сети: основные модели. – Воронеж: Изд-во ВГУ, 1999. – 157 с.;
* Боголюбов, А.Н. О вещественных резонансах в волноводе с неоднородным заполнением / А.Н. Боголюбов, А.Л. Делицын, М.Д. Малых // Вестник ЮУрГУ. Серия «Математика, физика, химия». – 2001. – Вып. 2. – № 5(14). – С. 23–25.

*Пример 2.*

* база данных Detexify (http://detexify.kirelabs.org), база MNIST;
* синтаксис LaTeX, MathCad®, MathML, Wolfram Mathematica®;
* данные конкурса CROHME в виде inkML-файлов.

1. **Перечень подлежащих разработке вопросов:**

*Пример.*

* рассмотрение существующих научных проектов, генерирующих потоки данных и вычислительных задач с различной степенью интенсивности;
* анализ современных программных технологий для организации масштабируемых вычислений;
* разработка собственного программного комплекса для организации масштабируемых научных вычислений;
* оценка работоспособности разработанного программного комплекса в различных режимах и внешних условиях.

1. **Дата выдачи задания:** 3 февраля 2022 г.

Руководитель практики со стороны ЮУрГУ:

Доцент каф. ЭВМ, к.т.н.              \_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И.О.

должность, ученая степень подпись ФИО ответственного

Руководитель практики со стороны предприятия:

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

должность, ученая степень подпись ФИО ответственного

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*И.О. Фамилия* /

КАЛЕНДАРНЫЙ ПЛАН

| Этап | Срок сдачи | Подпись  руководителя |
| --- | --- | --- |
| Введение | 04.02.2022 |  |
| Обзор литературы | 11.02.2022 |  |
| Разработка модели, проектирование | 18.02.2022 |  |
| Реализация системы | 25.02.2022 |  |

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*И.О. Фамилия*/

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*И.О. Фамилия*/

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc31816740)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 10](#_Toc31816741)

[1.1. ОБЗОР АНАЛОГОВ 10](#_Toc31816742)

[1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ 13](#_Toc31816743)

[1.3. ВЫВОД 14](#_Toc31816744)

[2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ 15](#_Toc31816745)

[2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 16](#_Toc31816746)

[2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 18](#_Toc31816747)

[3. ПРОЕКТИРОВАНИЕ 20](#_Toc31816748)

[3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ 20](#_Toc31816749)

[3.2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ 22](#_Toc31816750)

[3.3. ОПИСАНИЕ ДАННЫХ 24](#_Toc31816751)

[4. РЕАЛИЗАЦИЯ 26](#_Toc31816752)

[4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ 26](#_Toc31816753)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 28](#_Toc31816754)

[ПРИЛОЖЕНИЕ А ФРАГМЕНТ ИСХОДНОГО КОДА 30](#_Toc31816755)

# ВВЕДЕНИЕ

Во *введении* отражаются следующие основные моменты:

* общая формулировка темы;
* актуальность выбранной темы, её теоретическое и/или практическое значение;
* степень научной проработанности темы исследования;
* цель и задачи исследования;
* объяснение того, как, с помощью каких методов исследования и в каком порядке автор намеревается решать поставленные задачи;
* анонс структуры работы (названия глав работы и их краткая характеристика);
* характеристика основных источников информации.

Введение должно быть кратким (обычно до 3 страниц) и четким. Из введения должно быть понятно, чему посвящена работа (цель работы), какие задачи и с помощью каких методов в ней решаются, какие результаты должны быть достигнуты.

*Проблема исследования* – теоретический или практический вопрос, требующий исследовательского изучения и решения средствами науки. Постановка научной проблемы – творческий акт, требующий особого видения, специальных знаний, опыта, научной квалификации. Решение проблемы обычно составляют цель следования.

*Тема* – лаконичная формулировка проблемы. Наиболее убедительным основанием, определяющим тему исследования, является противоречие в социальной практике, практике менеджмента, отражающее самые острые, общественно значимые вопросы, требующие безотлагательного решения.

Критерий *актуальности* указывает на необходимость и своевременность изучения и решения обозначенной проблемы. Актуальные исследования дают ответ на наиболее острые в данное время вопросы, отражающие социальный заказ общества, бизнеса современной науке, указывают на важнейшие противоречия, которые имеют место в практике. Критерий актуальности динамичен, подвижен, зависит от времени и учета конкретных и специфических обстоятельств. В самом общем виде актуальность характеризуется степенью расхождения между спросом на научные идеи и практические рекомендации (для удовлетворения той или иной потребности) и предложениями, которые могут дать наука и практика в настоящее время.

*Пример.*

*Актуальность исследовательской темы*, связанной с исследованиями в области масштабируемых научных вычислений в гетерогенных средах заключается в том, что модель распределенных вычислений, в частности, облачные технологии и ГРИД стали *особо важной частью* научного мира, можно даже сказать – ключевой. В течение последних десятилетий произошла интенсивная стандартизация и глобализация использования всех видов компьютерных ресурсов, вызванная *ростом сложности решаемых задач*. Более того, функционирование многих фундаментальных исследовательских проектов стало невозможным в принципе без высокопроизводительных вычислительных систем. Ясно, что фундаментальные и прикладные исследования в области мультимасштабного компьютинга могут привести к качественным изменениям, как в научном мире, так и в области коммерческих решений.

Анализ *степени разработанности проблемы* демонстрирует ее проработанность и изученность в соответствующей науке и практике.

*Цель исследования* – это мысленное предвосхищение результата, который будет получен в ходе исследования (каким его видит исследователь). Цель считается достигнутой, если сформулирована, обоснована, доказана и проверена на практике ведущая идея, отраженная в теме.

*Пример.*

Целью представленной работы является разработка программного комплекса обеспечивающего объединение гетерогенных распределенных ресурсов с использованием ПО BOINC, SLURM и GlusterFS.

*Задачи исследования* – это пути достижения цели (что нужно сделать, чтобы цель была достигнута: изучить, описать, установить, выявить и т.д.). Если для решения проблемы нужен теоретический анализ литературы, значит, одной из задач может быть выявление теоретических основ проблемы и т.д. Формулировать задачи нужно очень тщательно, поскольку описание их решения должно составлять содержание глав и параграфов исследования, а от описания их решения будет зависеть оценка результативности исследования.

*Пример.*

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

1. Рассмотреть существующие научные проекты, генерирующие потоки данных и вычислительных задач с различной степенью интенсивности.
2. Провести детальный анализ современных программных технологий для организации масштабируемых вычислений в контексте гетерогенных вычислительных сред.
3. Разработать архитектуру собственного программного комплекса на базе существующих аппаратных решений и представить конечную её реализацию с использованием ПО BOINC, GlusterFS и SLURM.
4. Оценить работоспособность разработанного программного комплекса в различных режимах и внешних условиях (интенсивность потоков данных и задач, инфраструктурная конфигурация).
5. Произвести выборку результатов и проанализировать работу системы, в домене специально разработанных задач.

*Методы исследования* – это способы решения научно-исследовательских задач и получения результата исследования. Обычно используются теоретические методы (анализ, синтез, сравнение, обобщение, моделирование и т.д.), эмпирические методы, обеспечивающие сбор данных (наблюдение, изучение продуктов деятельности, документации, анкетирование, социометрия, беседа, метод независимых характеристик, эксперимент и т.д.), математические методы (обработка количественных данных, ранжирование и т.д.).

*Новизна исследования* характеризует новые теоретические и практические выводы, закономерности, содержание, принципы и технологии, которые к данному моменту не были известны и не зафиксированы в литературе. Критерий новизны исследования может иметь как теоретическое, так и практическое значение. Теоретическое значение исследования заключается в создании концепции, описании метода, модели, подхода, понятия, принципа, и т.д. Практическая значимость исследования состоит в его готовности к внедрению в практику.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В данном разделе описывается предметная область, постановка задачи дается в развернутом виде в соответствии с требованиями заказчика, но сформулированными с профессиональной помощью студента, могут быть приведены имеющиеся аналоги и охарактеризованы выбранные для достижения цели программные средства. Вся приведенная здесь информация может быть описана в одном разделе или разбита на нижеприведенные подразделы.

Задача может относиться к очень сложной предметной области, с новыми понятиями, со сложной терминологией, сложными взаимосвязями между объектами, поэтому необходимым является описание этой предметной области, решенных, нерешенных или частично решенных проблем, краткой истории развития, вкладом предшественников, описанием понятий и терминов, с которыми познакомился студент.

Описание предметной области должно затрагивать более широкий и важный круг проблем, чем у решаемой в работе задачи. Например, если решается задача разработки и создания микроконтроллера с программной компонентой, то имеет смысл описать область микроэлектроники.

## 1.1. ОБЗОР АНАЛОГОВ

В этом разделе необходимо описать известные аналоги разрабатываемых студентом средств. Если таких работ много, то необходимо предоставить информацию о самых используемых и качественных, на данный момент времени, ПС или ПТС. Аналоги могут быть не для всего ПС или ПТС, а только какой-то его части, например, только технической части или программной компоненты. Пример ссылки на библиографические источники: [1-18].

*Пример.*

На сегодняшний день самой крупной, межгосударственной ГРИД сетью оперирует проект WLCG (Worldwide LHC Computing GRID) [7-9], организованный в ЦЕРНе.

WLCG сеть состоит из четырех уровней, называемых “Tiers” (ярус, уровень). Всего существует 4 уровня, соответственно 0, 1, 2, 3. Каждый уровень организован группой компьютеров и вычислительных центров. При том, с ростом индекса уровня вычислительные возможности предполагаются более слабыми с точки зрения вычислительных емкостей по отношению к предыдущим [7, 8].

Вычислительные центры Tier 1 (рисунок 1) представлены крупными вычислительными узлами.

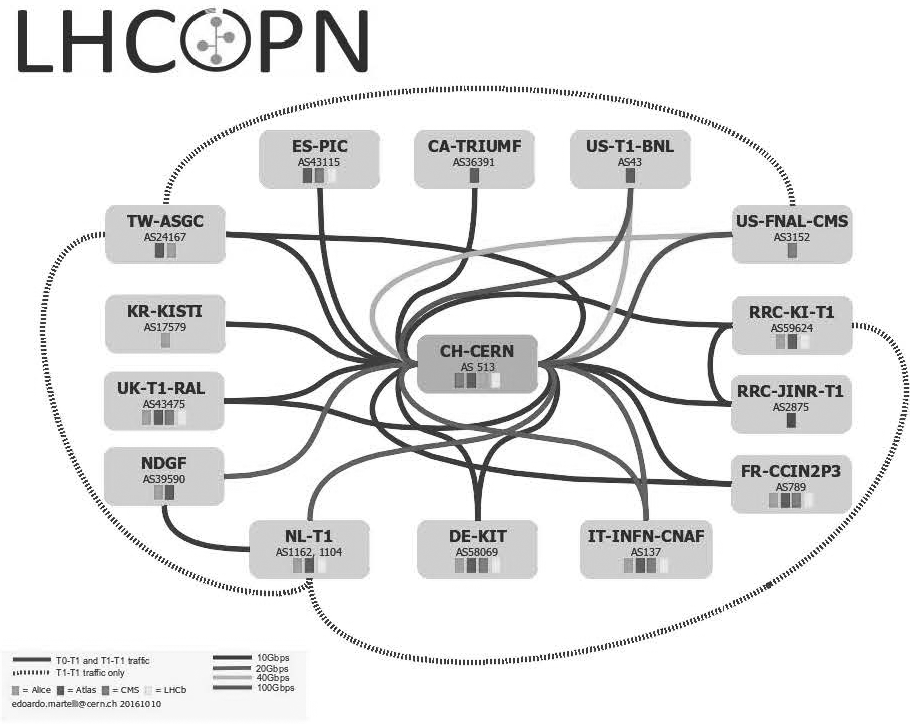


Рисунок 1 – Архитектура Worldwide LHC Computing Grid

Файловая система и в частности, распределенная файловая система, является очень важной составляющей любого вычислительного эксперимента [16-18]. Являясь частью операционной системы, она предоставляет необходимый инструментарий для манипулирования устройствами постоянного хранения данных, позволяя абстрагироваться от конкретной природы физических процессов на аппаратном уровне.

Распределенная файловая система должна быть прозрачной (от англ. transparency), отказоустойчивой и масштабируемой [18]:

* *прозрачность* предполагает обеспечение надлежащего уровня абстракции для конечного пользователя;
* *устойчивость* подразумевает способность системы адекватно отрабатывать различные сбои, к примеру, в работе сетевого оборудования;
* *масштабируемость* подразумевает способность системы оперировать в условиях переменного количества узлов без существенных изменений в качестве обслуживания.

На данный момент, разработано большое количество свободно-распространяемых распределенных файловых систем. Тем не менее, ни одна файловая система из таблицы 1 не является абсолютно универсальной. Большинство распространенных реализаций адаптированы и оптимизированы для вполне определенного класса задач.

Таблица 1 – Сравнительный анализ областей применения распределенных файловых систем [18]

| **Область применения** | **Файловая система (DFS)** |
| --- | --- |
| Большие данные | HDFS[17, 18]  MapR FS |
| Суперкомпьютинг | Lustre [17, 18]  Panasas [18] |

Продолжение таблицы 1

| **Область применения** | **Файловая система (DFS)** |
| --- | --- |
| Многоцелевые | (p)NFS  Ceph [17, 18]  Gluster-FS [17, 18]  XtreemFS  MooseFS  iRODS |
| Специфико-ориентированные | dCache [18]  XRootD  CernVM-FS [18]  EOS |

## 1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

В этом разделе необходимо описать известные технические средства, программные средства, технологии, которые выбирали для решения данной задачи. Описать достоинства и недостатки используемых средств и обосновать необходимость использования именно этих программных сред, средств и технологий. В частности, такими достоинствами могут являться низкая цена программного или технического средства или бесплатность программного обеспечения, наличие графического интерфейса и т. п.

*Пример.*

Для проведения исследований в рамках работы была выбрана файловая система – GlusterFS [16, 18]. Она позиционируется разработчиками как высокомасшабируемая децентрализованная файловая система. Важным преимуществом GlusterFS является то, что метаданные не хранятся в едином месте, более того отсутствует сервер метаданных (таблица 2).

Таблица 2 – Сравнительный анализ распределенных файловых систем [18]

|  | **HDFS** | **iRODS** | **Ceph** | **GlusterFS** | **Lustre** |
| --- | --- | --- | --- | --- | --- |
| Architecture | Centralized | Centralized | Distributed | Distributed | Centralized |
| Naming | Index | Database | CRUSH | EHA | Index |
| API | CLI, FUSE, REST, API | CLI, FUSE, API | FUSE, mount,  REST | FUSE, mount | FUSE |

## 1.3. ВЫВОД

Здесь четко и ясно должно быть сформулирована суть решаемой задачи: является ли задача частью сложной большой системы или это автономная задача. Как правило, задачей студента является разработка и создание программного средства или технического средства с программной компонентой для решения задач физики, техники, биологии, генетики, геофизики, археологии и т. п. Задача в этом случае может состоять из следующих составляющих:

* проблемы предметной области, например, задачи создания структуры данных для более удобного и быстрого поиска информации, технической проблемы и т. п.;
* задачи по проектированию и созданию соответствующего программного средства.

# 2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

При постановке задачи требования к ПС или ПТС могут быть следующие:

* общие или бизнес-правила;
* функциональные;
* нефункциональные (например, качества и надежности, требования к базовому аппаратно-программному обеспечению).

К *общим**требованиям*(бизнес-правилам) относятся требования по использованию бесплатных инструментов, использованию специфических или устаревших технических или программных средств, повышенные требования к конфиденциальности использования информации и т. п.

*Пример.*

Для реализации данной системы необходим следующий набор подсистем:

1. *HTTP-сервер Apache* с веб-интерфейсом в виде набора *РНР-скриптов*. С помощью веб-сайта обеспечивается общее управление системой, создание, конфигурация, управление и получение результатов заданий и регистрация новых пользователей.
2. *MySQL* база данных, обеспечивающая хранение данных о пользователях, заданиях.
3. *Набор сервисов* (генератор заданий, файловая система, планировщик, валидатор, ассимилятор результатов), обеспечивающих организацию взаимодействия с пользователем и данными, выдача, получение ассимиляция результатов.
4. *Клиентское приложение под различные платформы* – организация выполнения вычислений за счет ресурсов пользователя, получение задания от сервера и отправка результатов на сервер.
5. *Графический интерфейс клиентского приложения* – настройка и контроль работы клиентского приложения, визуальное отображение регистрации в системе, получения задания и прогресса их выполнения;
6. *Графический интерфейс администратора* – настройка и контроль работы вычислительной системы.

## 2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Одним из самых важных моментов при постановке задачи является определение *функциональных требований*, т. е. функций, необходимых пользователю, которые нужно реализовать для достижения цели. Пользователей можно разбить на типы (если есть в этом необходимость) и перечислить функции в соответствии с их предназначением для каждого типа. Если речь идет о требованиях к ПТС, то это могут быть требования, как к программной, так и к технической компоненте.

Эти требования можно оформить в виде коротких и четких предложений, описывающих функции ПС или ПТС, либо в виде диаграмм прецедентов (Use case diagram) языка UML.

*Пример.*

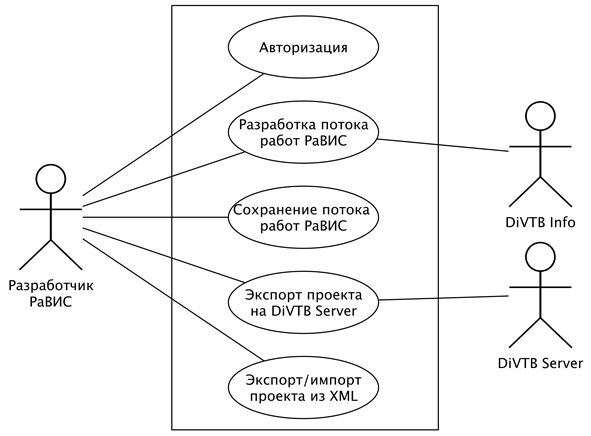


Рисунок 2 – Диаграмма вариантов использования системы DiVTB Developer

На рисунке 2 изображена диаграмма вариантов использования системы DiVTB Developer. Можно выделить следующих ключевых актеров, взаимодействующих с системой:

1. *Разработчик РаВИС*:как правило, прикладной программист, создающий распределенный виртуальный испытательный стенд для инженера путем описания необходимых для расчета этапов моделирования и их связей.
2. *DiVTB Info*: подсистема DiVTB, предоставляющая другим подсистемам информацию об имеющихся в распределенной вычислительной среде доступных сервисах (установленных на узлах РВС пакетах инженерного моделирования), имеющихся лицензиях на CAE-пакеты, а также о параметрах каждого сервиса.
3. *DiVTB Server*: подсистема DiVTB, выполняющая задачи управления виртуальными экспериментами, включая загрузку, запуск и исполнение виртуальных экспериментов на основе загруженных РаВИС, отслеживание статуса запущенных экспериментов и получение результатов их выполнения.

Рассмотрим прецеденты использования разрабатываемой системы.

Пользователю необходимо *авторизоваться*, чтобы иметь возможность работать в системе DiVTB Developer: для этого он вводит в соответствующем интерфейсе системе свои имя учетной записи и пароль.

После процедуры авторизации пользователь системы может начать *разработку РаВИС* путем описания потока работ.

Разрабатываемый пользователем поток работ РаВИС сохраняется и затем может быть заново открыт для редактирования.

Пользователь может *импортировать* поток работ РаВИС в разрабатываемую систему, загрузив *XML-файлы* его описания, а также экспортировать разработанный в системе поток работ в формате XML.

Кроме того, пользователь может *экспортировать поток работ* на DiVTB Server, откуда возможен запуск расчета задачи в распределенной вычислительной среде.

## 2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

*Нефункциональные требования*определяют условия и среду выполнения функций (например, защита и доступ к БД, секретность и др.), они непосредственно не связаны с функциями, а отражают пользовательские потребности к выполнению функций. Они характеризуют принципы взаимодействия со средами или другими системами, а также учитывают время работы, защиту данных, а также стандарты качества для достижения отдельных показателей или атрибутов качества. Эти требования отражают потребности заказчиков системы. Например, требования качества и надежности ПС или ПТС могут быть сформулированы заказчиком работ (руководителем).

Необходимо помнить, что любое дополнительное требование отразится на проекте ПС или ПТС, затратах и времени реализации.

*Пример.*

В системе должна быть предусмотрена защита от следующих типов угроз безопасности:

1. Подделка результатов вычислений.
2. Подделка времени выполнения.
3. Заражение вирусом результата вычислений.
4. Заражение вирусом распространяемого задания.
5. Умышленное переполнение базы данных сервера.
6. Кража данных учетной записи с сервера.
7. Перехвата передаваемых данных учетной записи.
8. Кража файлов проекта.
9. Кража файлов участников сети.
10. Злоупотребление ресурсами участников сети.

Проблемы в пунктах 3,4 решаются путем генерирования цифровых подписей для исполняемых модулей.

Проблемы в пунктах 1,2,5,6,8,9,10 решаются с помощью анализа журналов системы контроля версий, поиска уязвимостей и установки stable релизов ПО и ядра ОС.

Проблемы в пункте 7 решаются путем шифрования SSL информационного потока (RSA, DSA, AES, DES).

# 3. ПРОЕКТИРОВАНИЕ

В этом разделе необходимо описать методы реализации ПС или ПТС (со схемами, в частности, структурной, функциональной, принципиальной схемой, схемой соединений, подключений, общей, расположения), используемые математические алгоритмы решения задачи (если они есть) и исследовательскую часть, если она необходима для решения проблемы. Вся информация может быть описана как в одном разделе, так и разбита, например, на нижеприведенные подразделы.

## 3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

В этом разделе может быть полностью или частично описана структура ПС или ПТС, т. е.:

* из каких функциональных блоков (файлов, модулей, процедур, функций, классов) состоит ПС или ПТС;
* приведено описание каждого блока с его названием и назначением;
* приведена графическая схема взаимосвязи этих блоков.

Для описания схемы функционирования программного средства можно использовать диаграммы UML.

*Пример.*

При описании архитектуры разрабатываемой игровой платформы была спроектирована диаграмма компонентов для отображения взаимодействия логических частей приложения, представленная (рисунок 3).

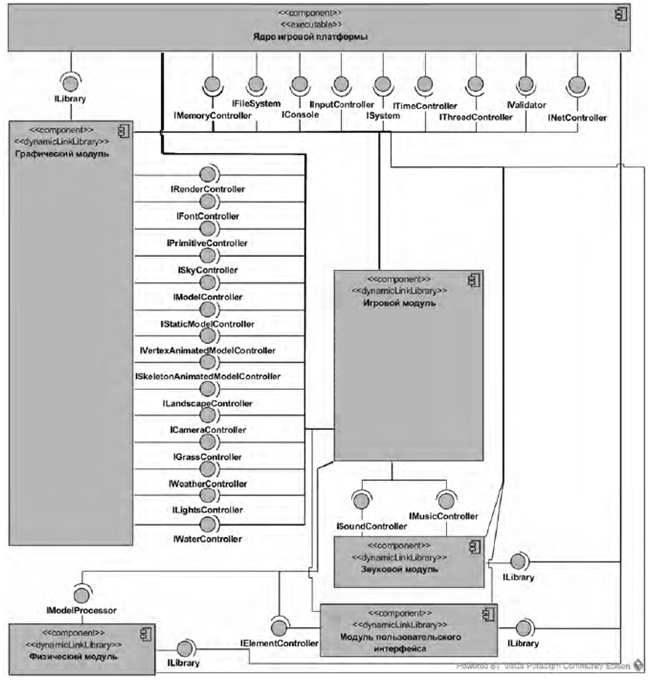


Рисунок 3 – Диаграмма компонентов проектируемой системы

## 3.2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

*Исследовательская часть.* Исследовательская часть заключается в изучении и сравнении имеющихся методов решения задачи, какие достоинства и недостатки есть в каждом из способов, как использовать достоинства и избежать недостатков в предлагаемом студентом методе решения задачи.

*Математические алгоритмы.* В этом подразделе необходимо описать используемые математические алгоритмы решения задачи с формулами и со ссылками на источники.

*Пример.*

Фазу сигнала, рад, вычисляют по формуле (1)

|  |  |
| --- | --- |
| , | (1) |

где  – частота сигнала, рад/с;

 – время, с.

*Алгоритмы реализации ПС или ПТС.* Для описания алгоритма реализации ПС или ПТС могут использоваться два способа описания алгоритмов:

* словесный: в виде последовательности шагов по реализации с описанием классов, интерфейсов, процедур (можно с программным кодом, реализующим наиболее интересные алгоритмы), а если это техническая часть, то помимо словесного описания могут быть приведены схемы технических устройств;
* графический: в виде блок-схем обязательно с пояснениями.

*Пример*.

Для решения проблемы был разработан алгоритм многопроходной визуализации сцены (рисунок 4).

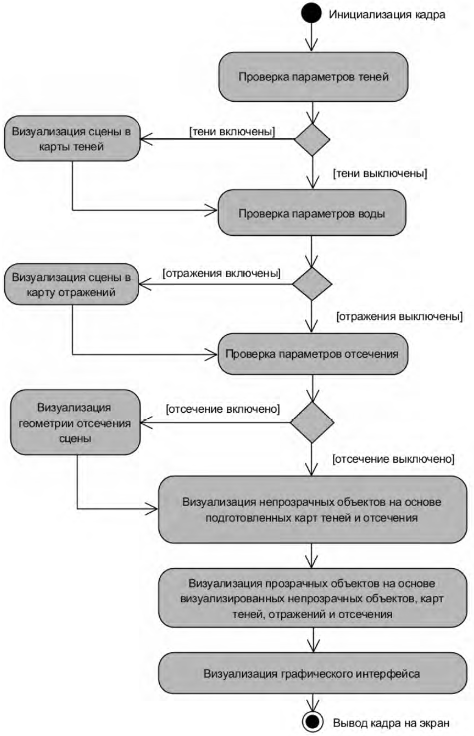


Рисунок 4 – Диаграмма алгоритма формирования кадра

## 3.3. ОПИСАНИЕ ДАННЫХ

В данном разделе описывается структура входных, выходных и промежуточных данных. Например, входные данные поступают на вход программы в виде файла, значит, структура файла должна быть полностью описана. Аналогичное требование выполняется для промежуточных и выходных данных. Например, для сайтов входной информацией могут быть текстовые файлы (указать в каком формате), графическая информация (указать формат) и т. п. Выходная информация – это HTML, PHP и т. п. страницы, видимые в окне браузеров (указать, каких).

*Описание базы данных.* В этом разделе необходимо описать структуру базы данных, если таковая имеется в проекте. Если приложение не использует базу данных, этот пункт опускается.

При описании структуры базы данных:

* описываются все таблицы в виде: *имя таблицы*и её назначение, т.е. для хранения какой информации предназначена данная таблица;
* описание всех полей таблицы с указанием типа, назначения, первичных и внешних ключей.

Для объектной базы данных приводится описание структуры данных всех классов, аналогично тому, как это сделано для таблиц, добавляется только описание методов.

Если в работе разработаны и созданы инструменты для работы с БД, их можно описать в данном разделе.

*Пример.*

Получаем базу данных (рисунок 5).

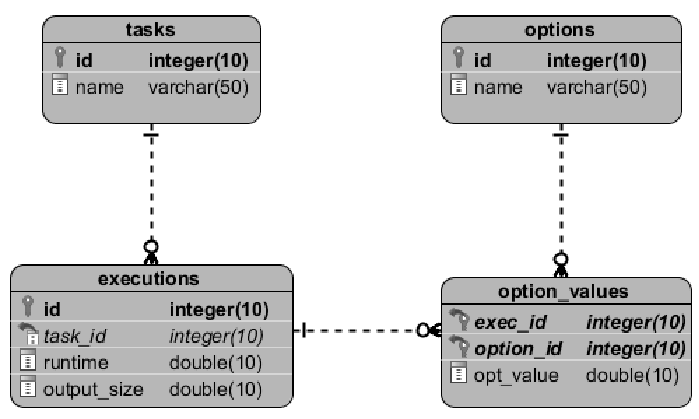


Рисунок 5 – Схема разработанной базы данных истории запусков

# 4. РЕАЛИЗАЦИЯ

## 4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

В этом разделе демонстрируются и анализируются основные достижения, полученные в результате выполнения работы. Примеры работы программы при различных входных данных, можно привести сводные таблицы и графики. Если основной задачей работы было провести исследование, то приводятся методы и результаты исследования, классификация результатов, табличные сравнительные данные.

В этом разделе можно привести снимки экранов с пояснениями для лучшего восприятия раздела или фотографии технической системы.

*Пример.*

Фотография разработанной системы (рисунок 6).



Рисунок 6 – «Сердце» разработанной системы

После сборки из исходного кода/загрузки через пакетный менеджер BOINC клиента пользователь может выбрать, в каких проектах ему принять участие и присоединиться к выбранным (рисунок 7), копируя URL адрес в меню BOINC Manager.

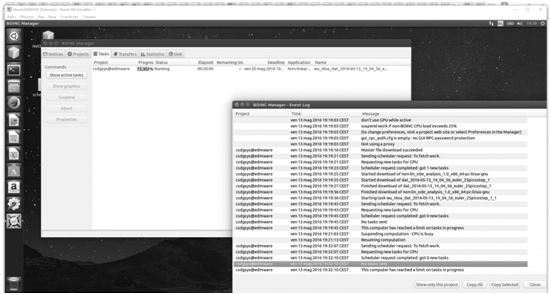


Рисунок 7 – Инициализация окружения в рабочем цикле (Linux GUI)

После внедрения системы путем незначительного усложнения этапов подготовки задания удалось автоматизировать функции, связанные с распространением, сбором и агрегированием результатов, созданием отчета.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Заенцев, И.В. Нейронные сети: основные модели. – Воронеж: Изд-во ВГУ, 1999. – 157 с.
2. Kitainik L. Fuzzy Decision Procedures with Binary Relations. Towards a Unified Theory / L. Kitainik. – Boston: Kluwer, 1967. – 254 p.
3. Бражник, А.А. Справочное пособие по бухгалтерскому учету в промышленности / А.А. Бражник, Б.В. Щеголов. – Минск: Белорусь, 1969. – 488 с.
4. Круглов, В.В. Нечеткая логика и искусственные нейронные сети: учебное пособие / В.В. Круглов, М.И. Дли, Р.Ю. Голунов. – М.: Физматлит, 2001. – 224 с.
5. Электробезопасность на открытых горных работах: справ. пособие / В.И. Щупкий, А.И. Сидоров, Ю.В. Ситчихин, НА. Бендяк. – М.: Недра, 1996. – 266 с.
6. Модели принятия решений на основе лингвистической переменной / А.Н. Борисов, А.В. Алексеев, О.А. Крумберг и др. – Рига: Зинатне, 1982. – 256 с.
7. Металлические конструкции: учебник: в 3 т. / под ред. В.В. Горева, – 2-е изд., перераб. и доп. – М.: Высшая школа, 2001. – Т. 1. – 551 с.
8. Методические указания по разработке вопросов охраны труда в дипломных проектах студентов приборостроительного факультета (специальности 0647, 0608) / Г.И. Матвеев; под ред. А.В. Хашковского. – Челябинск: ЧПИ, 1982. – 18 с.
9. Сетевые методы планирования и управления: методические указания / сост. В.С. Зинкевич, Л.А. Баев, Н.П. Мешковой. – Челябинск: Изд-во ЮУрГУ, 1998. – 22 с.
10. Антонов, В.М. Использование обобщённых ситуаций при формировании памяти робота с обучаемой системой управления / В.М. Антонов // Теория и техника автоматического управления: сб. науч. тр. – Томск: УНПК «Кибернетика» Томского политехн. ин-та, 1990. – С. 172 – 181.
11. Боголюбов, А.Н. О вещественных резонансах в волноводе с неоднородным заполнением / А.Н. Боголюбов, А.Л. Делицын, М.Д. Малых // Вестник ЮУрГУ. Серия «Математика, физика, химия». – 2001. – Вып. 2. – № 5(14). – С. 23–25.
12. Bandler, W. Fuzzy power sets and fuzzy implication operators / W. Bandler, L. Kohout // Fuzzy Sets and Systems. – 1980. – V. 4. – P. 13 – 30.
13. Вишняков, И.В. Модели и методы оценки коммерческих банков в условиях неопределенности: автореферат дис. ... д-ра экон. наук / И.В. Вишняков. – М.: Изд-во МГУ, 2002. – 34 с.
14. Пат. 2187888 Российская Федерация, МПК7 Н 04 В 1/38, Н 04 .У 13/00. Приемопе-редающее устройство / В.И. Чугаева. – № 2000131736/09; заявл. 18.12.00; опубл. 20.08.02, Бюл. № 23 (II ч.). – 3 с.
15. ГОСТ 12.1.005–75. ССБТ. Воздух рабочей зоны. Общие санитарно-гигиенические требования. – М.: Изд-во стандартов, 1975. – 55 с.
16. Тархов, Д.А. Нейрокомпьютеры и их применение. В 20 кн. Кн. 18: Нейронные сети. Модели и алгоритмы / Д.А. Тархов. – М.: Радиотехника, 2005. – 253 с.
17. Осовский, С. Нейронные сети для обработки информации / С. Осовский; пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.
18. Клуб программистов. – http://www.programmersclub.ru. Дата обращения: 15.02.2022.

# ПРИЛОЖЕНИЕ А ФРАГМЕНТ ИСХОДНОГО КОДА

*Пример.*

#!/bin/bash

# Generate 20 files with 10 rows

nflies = 25

nrows = 10

for ((i = 0; i<$nflies; i++)); do

filename = "dat\_"$(date + %F\_%H\_%M\_%S)"\_micst\_"$nrows"ent\_perdata\_"$i

echo - e $i": Next "$nrows" rows ... \n"

for ((j = 0; j<$nrows; j++)); do

echo - e $(shuf - i 0 - 2 - n 1)"."$(shuf - i 0 - 999 - n 1)" 1.000 1.000 1.000 0.000 3.000 0.000025\n" >> . / workunits / $filename

done

done

for fname in $(ls ./ workunits); do

cp ./ workunits / $fname `./ bin / dir\_hier\_path $fname`

#rm - f ./ workunits / $fname

# --target\_user 8

./bin/create\_work --appname non - lin\_ode\_analysis --min\_quorum 1 --target\_nresults 1 --wu\_name wu\_nloa\_$fname --wu\_template templates/default\_in --result\_template templates/default\_out $fname

done

rm - f . / workunits/\*

9