

№	Краткое описание	Речь
1	Тема, ФИО	Уважаемые члены Государственной Аттестационной комиссии, хочу предложить вашему вниманию вторую часть дипломного проекта, первую часть которого только что представил мой коллега.
2	Консоль	<p>В первой части было представлено ядро системы. Оно по идеологическим причинам работает в консольном режиме. Этот режим очень удобен для запуска и функционирования, но при этом он не очень удобен для восприятия человеком.</p> <p>Для управления консольными приложениями часто используют конфигурационные файлы. Но у этого способа есть ряд недостатков, обусловленные сложностью управления и возможными ошибками.</p>
3	Структура проекта - страшная схема	Ядру, как и любой другой системе, необходимо управление и для этого мною была разработана Панель управления – инструмент управления работой исполняемой среды и визуализации информации процесса и результатов этой работы. Также был разработан Прикладной интерфейс программирования , который позволяет разрабатывать модули мониторинга.
4	Модель приложения MVC	<p>В основу модели приложения Панели управления был положен шаблон проектирования, называемый MVC или Model-View-Controller. Основная идея этого шаблона – разделение приложения на три основных компонента со «слабой» связанностью между ними.</p> <p>Компонент Модель реализован классом домен, который содержит информацию о ядре и логику получения, обработки и предоставления этой информации.</p> <p>Компонент Представление или интерфейс это класс Главное окно, которое является пользовательским интерфейсом в классическом понимании.</p> <p>Компонентом Контроллер является класс координатор, который согласовывает информацию из домена и ее отображение в пользовательском интерфейсе.</p>
5	Диаграмма последовательностей	На слайде представлена общая схема функционирования Панели управления и ее взаимодействия с ядром. Ядро через драйвер Исследователь оповещает панель о своей активности через определенные промежутки времени. Так же с помощью этого вызова передается минимальная информация о ядре, на основе которого делается вывод об изменении ядра. Если изменение произошло, то домен обновляет локальную информацию и оповещает об этом интерфейс, который обновит отображение. Если же пользователь совершит какое-либо действие, то через систему обработки событий об этом станет известно координатору, который организует соответствующие изменения в домене. После изменений в ядре информация в домене обновляется, о чем оповещается интерфейс.
6	Архитектура приложения диаграмма классов	Архитектура приложения включает в себя множество классов, которые условно можно отнести к трем основным компонентам, соответствующим структуре паттерна MVC. Каждый компонент MVC реализован совокупностью классов.

№	Краткое описание	Речь
		<p>К модели можно отнести такие основные классы как Domain — контейнер для информации о ядре, DomainController — обеспечивает инициализацию Домена, Discoverer и DiscovererAdapter — отвечают за оповещение панели об активности ядра.</p> <p>К контроллеру относятся классы: Координатор — который соотносит манипуляции пользователя и вызовы методов домена, классы слушателей ActionListener — классы содержащие логику взаимодействия.</p> <p>Интерфейс представлен набором классов окон приложения. MainFrame — является главным окном и с ним производится большая часть манипуляций. Другие классы — это InternalFrame внутреннее окна, которые являются дочерними по отношению к главному окну.</p>
7	Хранение информации Домен	<p>В терминах нашей системы контекст ядра — это минимальная информация о ядре и отдельных узлах. Эта информация хранится в динамических массивах и списках, которые в свою очередь содержатся в контейнере Domain. Информация кэшируется и если она требуется, то берется из этого кэша, а не запрашивается у ядра каждый раз за исключением ситуаций, когда необходимо принудительно обновить информацию. Такой подход позволяет снизить уровень трафика сетевого взаимодействия и повысить отзывчивость панели управления. Домен содержит адаптер Исследователь, который предоставляет ядру интерфейс, через который контекст доставляется панели. Поскольку в домене хранится только минимальная информация, то вся дополнительная получается через интрейфейсы.</p>
8	Взаимодействие компонентов – Координатор	<p>Основной задачей координатора является установление соответствия между действиями пользователя и изменением информации в домене. Таким образом получается, что взаимодействие Домена и Интерфейса выглядит прозрачно.</p>
9	Визуализация – Главное окно	<p>В качестве основы для графического интерфейса была взята концепция многодокументного интерфейса (MDI). Такая организация интерфейса очень удобна, если необходимо работать с большим количеством однотипных окн.</p> <p>Панель управления содержит информацию о большом количестве узлов. Так как на узле функционирует несколько модулей, то для отображения информации о модулях потребуется много открытых окн.</p>
10	Использование API кусочек кода	<p>Одной из ключевых особенностей панели управления является прикрепление модулей к узлам. Для удобства и универсальности был разработан соответствующий прикладной интерфейс программирования или API. Прикладной интерфейс программирования позволяет разрабатывать модули мониторинга на основе унифицированного каркаса исходного кода модуля. Этот интерфейс задает правила исполнения, передачи параметров и возврата результата модуля. В текущей реализации интерфейс программирования модулей представляется каркасом с одним публичным методом – «invoke(..)». Параметры модуля мониторинга могут передаваться как обычная коллекция или список языка Python. Помимо</p>

№	Краткое описание	Речь
		интерфейсного метода invoke в модуле могут быть и любое количество внутренних методов.
11	Развертывание системы мониторинга	В итоге нами была разработана системы, для использования которой необходимо установить ядро системы на вычислительные узлы. Конфигурация ядра произойдет автоматически . Затем достаточно запустить Панель управления на ЛЮБОМ узле и система готова к работе с базовым функционалом. Если у пользователя возникнет необходимость в расширении функционала, то это можно сделать установкой модулей , удовлетворяющих интерфейсу API. Затем система будет выполнять возложенные на нее задачи и возвращать результат этого выполнения.
12	Сравнительная характеристика	На слайде представлена сравнительная характеристика разработанной нами системы с двумя другими программными продуктами, наиболее близкими по функциональным требованиям. Отличительной особенностью предлагаемого нами решения является сочетание в себе сильных сторон других приложений, но которые не предлагали такого комплексного решения.
13	Итоги	В процессе выполнения дипломной работы мною были спроектированы и реализованы подсистемы визуализации и управления. Так же был разработан интерфейс программирования модулей.
14	Спасибо за внимание!	