

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ  
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

---

Факультет Автоматики и Электроники  
Кафедра электрофизических установок

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту на тему:

Информационная система мониторинга и представления основных  
рабочих параметров импульсного источника нейтронов ИБР-2 в  
режиме реального времени.

Студент дипломник \_\_\_\_\_ (                    )  
Руководитель проекта \_\_\_\_\_ (                    )  
Рецензент \_\_\_\_\_ (                    )  
Зав. кафедрой \_\_\_\_\_ (                    )

Москва 2003

## Аннотация

В пояснительной записке представлено описание информационной системы, разработанной для мониторинга и представления основных рабочих параметров импульсного источника нейтронов ИБР-2. Данная система формирует график мощности источника от времени, диаграмму состояния нейтронных каналов, а так же представляет некоторые дополнительные расчетные параметры на основе результатов измерений средней мощности установки и регистрации состояния нейтронных каналов. Доступ к данным осуществляется через глобальную сеть интернет.

## Оглавление

Введение .....	4
Глава 1: Импульсный источник нейтронов ИБР-2 .....	6
Глава 2: Структура информационной системы .....	7
2.1 Измерительная система .....	9
2.2 Файловая система и сохранение данных .....	10
2.3 Формат файлов данных .....	11
2.4 Web-сервер .....	12
2.5 Приложение мониторинга и представления данных .....	12
Глава 3: Приложение мониторинга и представления данных .....	14
3.1 Выбор языка программирования для реализации .....	14
3.2 Представление данных .....	15
3.3 Компоненты приложения .....	17
3.4 Рабочий цикл программы .....	21
3.5 Файл конфигурации программы .....	26
3.6 Обработка ошибок .....	26
Глава 4. Результаты опытной эксплуатации приложения .....	28
Заключение .....	31
Литература .....	32
Приложение 1: Руководство пользователя информационной системы .....	33
1.1 Необходимое программное обеспечение .....	33
1.2 Выбор цикла для представления .....	33
1.3 Работа с программой представления .....	33
1.4 Ошибки при работе системы .....	34
Приложение 2: Руководство по установке и настройке программы .....	36
2.1 Необходимое программное обеспечение .....	36
2.2 Установка программного обеспечения .....	37
2.3 Настройка файла конфигурации .....	37

## Введение

Импульсный источник нейтронов ИБР-2 предназначен для обеспечения физических исследований на пучках нейтронов. Для обслуживания экспериментальных программ источник работает в течение 10 циклов в год, общее время работы источника за год составляет в среднем 2500 часов. Экспериментальные возможности импульсного источника нейтронов ИБР-2 доступны широкому научному сообществу.

Информационная система предоставляет обслуживающему персоналу источника, а так же исследователям, использующим пучки нейтронов для проведения экспериментов информацию о состоянии физической установки. Эта информация включает в себя мощность источника, развиваемую в данный момент, состояние (открытое, закрытое) каналов вывода пучков нейтронов, а также рассчитанные за период работы параметры: количество остановок источника, время простоя, и т.д.

Помимо информации о текущем состоянии экспериментальной установки, система сохраняет информацию о рабочих параметрах, полученных в ходе предыдущих циклов работы.

Данные доступны в виде автоматически генерируемых web-страниц содержащих следующую информацию:

- график изменения со временем средней мощности источника;
- диаграмму состояний транспортных каналов источника;
- численные измеренные и рассчитанные параметры.

Работа установки обычно производится циклами длительностью около 10 дней каждый. Информационная система предоставляет возможность просматривать как информацию о текущем цикле работы экспериментальной установки, так и сохраненную информацию о предыдущих циклах работы. Пользователь

системы имеет возможность получить детальную информацию о параметрах установки за выбранный им промежуток времени.

Далее представлено описание информационной системы в целом, а также подробное описание приложения мониторинга и представления данных, создание которого было выполнено в рамках работы над дипломным проектом.

## Глава 1: Импульсный источник нейтронов ИБР-2

Импульсный источник нейтронов ИБР-2 был введен в эксплуатацию в 1984 году. Название установки расшифровывается как *импульсный быстрый реактор*. Источник является базовой установкой лаборатории нейтронной физики ОИЯИ и включен в европейскую сеть ведущих нейтронных центров. На сегодняшний день реактор является одним из самых мощных экспериментальных импульсных источников нейтронов в мире. На реакторе учеными из 30 различных стран проводятся 250 экспериментов в год.

Усредненная мощность источника составляет 2 МВт; мощность в импульсе составляет 1500 МВт. Импульсы тепловых нейтронов длительностью 320 микросекунд испускаются с периодом в 5 герц. Относительно большая продолжительность вспышки тепловых нейтронов является характерной особенностью данной установки и выгодно отличает источник ИБР-2 от других импульсных источников.

Непосредственно вблизи реактора расположено несколько замедлителей. Все пространство вывода нейтронов разделено на 14 горизонтальных каналов. Первые 12 каналов оборудованы исследовательскими установками. Оставшиеся два канала используются для служебных целей. Каждый вывода канал оборудован специальным затвором, позволяющим перекрыть путь поступающим нейтронным импульсам. Открытие и закрытие любого затвора может производиться независимо.

Безопасность эксплуатации реактора осуществляется системой предотвращения аварий. В настоящее время проводится уточнение и улучшение характеристик реактора ИБР-2 с целью повышения эффективности экспериментальных исследований в области физики ядра и твердого тела.

## Глава 2: Структура информационной системы

Цель проекта – обеспечение доступа исследователей, использующих пучки нейтронов в своих экспериментах, к оперативной информации о состоянии импульсного источника и горизонтальных транспортных каналов вывода тепловых нейтронов. В целом перед информационной системой стоят следующие задачи:

- визуализация параметра средней мощности источника
- визуализация состояния транспортных каналов
- представление обобщенных рабочих параметров
- архивирование параметров и свободный доступ к ним

В состав информационной системы входят следующие основные компоненты:

- измерительная система
- сеть передачи данных
- сетевая файловая система
- программа мониторинга и представления данных
- web-сервер

Каждый компонент системы выполняет определенную задачу, при взаимодействии компонентов между собой используется технология «клиент-сервер». На иллюстрации 2.1 схематически представлена структура информационной системы в целом. Далее рассматриваются назначение и работа каждого компонента системы.



Рис. 2.1 Структура информационной системы

## 2.1 Измерительная система

Назначение измерительной системы – сбор данных о параметрах физической установки и их первичная обработка.

Основной метод, используемый для определения средней мощности реактора, состоит в измерении мощности каждого нейтронного импульса, испускаемого установкой. Мощность импульсов измеряется с помощью ионизационных камер, расположенных в каналах реактора. Ионизационные камеры функционируют в токовом режиме.

Диапазон измеряемой средней мощности находится в пределах от 0.02 МВт до 2.2 МВт. Относительная погрешность измерений мощности не превышает 1.5 процента. В дополнение к измерению мощности нейтронных импульсов производится измерение скорости потока охладителя установки и его температуры. Полученные в результате данные позволяют, рассчитав тепловую мощность, выделяемую реактором, контролировать адекватность получаемых значений средней мощности импульсного источника.

Помимо значений мощности, измерительная система регистрирует положение каждого из 12 затворов, расположенных в горизонтальных каналах, оборудованных исследовательскими установками. Данные о состоянии затворов поступают с пульта управления реактором и регистрируются измерительной системой асинхронно по мере изменения состояния затворов, т.е. каждое изменение состояния хотя бы одного затвора влечет за собой регистрацию данных. Для затвора каждого канала измерительная система регистрирует одно из следующих состояний:

- канал открыт
- изменение положения затвора
- канал закрыт

Измерительная станция создана на базе персонального компьютера. Использование персонального компьютера позволило использовать недорогие,

но эффективные средства аналогово-цифрового преобразования получаемых данных. В системе используется промышленная электроника фирмы Advantech и модули, изготовленные в лаборатории нейтронной физики специально для данной измерительной системы.

Измерительная система функционирует под управлением операционной системы MS-DOS. Данные принимаются и обрабатываются после каждого импульса реактора, т.е. приблизительно каждые 200 миллисекунд.

Каждые 30 минут выполняется вывод полученных параметров на экран измерительной станции. Если выполняется включение или выключение реактора или создается аварийная ситуация, связанная с работой реактора и требующая оперативного вмешательства, могут быть использованы специально предусмотренные алгоритмы обработки данных. Так, сотрудники установки могут получать не только значения средней мощности и состояния пучков нейтронов, но и некоторые дополнительные технологические параметры, такие как поток охладителя установки, его температуру и т.д. Измерительная станция расположена в непосредственной близости от пульта управления реактором.

## **2.2 Файловая система и сохранение данных**

Каждые 10 минут данные о средней мощности, развиваемой установкой, передаются по локальной вычислительной сети на файловый сервер для дальнейшего сохранения и обработки. Помимо этого, каждое изменение состояния любого из 12 затворов горизонтальных каналов инициирует передачу данных о новом состоянии.

Данные для каждого цикла работы реактора сохраняются в отдельном текстовом файле. При этом названия двух файлов, относящихся к одному циклу и содержащих информацию о мощности и состоянии каналов, формируются исходя из даты начала цикла, и отличаются расширениями. Например, файлы, содержащие результаты измерения рабочих параметров импульсного источника во время цикла, начало которого состоялось 8 апреля 2002 года, имеют следующие названия:

c08042002.pwr – файл данных о средней мощности,

c08042002.shb – файл данных о состоянии каналов

При этом число 08042002 является *идентификатором цикла* и в дальнейшем используется для указания на данный цикл при запросе пользователя системы.

Помимо указанных файлов, дополнительно формируется служебный файл с названием *status*, который содержит имя текущего цикла работы реактора и информацию о том, снимаются ли измерения в данный момент.

### 2.3 Формат файлов данных

Файл данных о средней мощности содержит дату и время, соответствующие моменту измерения с точностью до секунды, среднюю мощность источника, выраженную в киловаттах и относительную погрешность измерений. Каждая строка текстового файла содержит информацию об одном измерении.

Например, информация об измерении, сделанном 13 марта 2002 года в 9 часов, 50 минут и 37 секунд выглядит так:

```
13-03-2002 09:50:37 1144 0.8
```

Файл данных о состоянии каналов содержит дату и время, соответствующие моменту изменения состояния затворов и информацию о состоянии 12 затворов нейтронных каналов, используемых в исследовательских целях. Каждая строка текстового файла, как и в предыдущем случае, содержит информацию об одном измерении.

Например, информация о состоянии каналов на 13 марта 2002 года, 9 часов, 51 минуту и 55 секунд выглядит так:

```
13-03-2002 09:51:55 2 3 2 3 3 3 3 3 3 3 3 3
```

В среднем за цикл работы реактора сохраняется немногим более 50 тысяч результатов измерений. Все файлы содержатся в отдельной директории, в дальнейшем именуемой *рабочая директория*.

## 2.4 Web-сервер

Сервер принимает запрос от пользователя, декодирует его и выполняет программу обработки и представления данных, при этом программе передаются параметры запроса. Результат работы программы передается пользователю. В качестве web-сервера используется сервер Tomcat 3.2, работающий под управлением ОС Solaris 2.7.

## 2.5 Приложение мониторинга и представления данных

Задачей приложения мониторинга и представления данных является обработка результатов измерений и их подготовка к представлению пользователю.

Приложение формирует *индексную страницу*, которая содержит список доступных для представления циклов, и *страницу представления данных*, которая содержит параметры физической установки.

Данные об изменении во времени мощности источника представляются в виде графика. Данные об изменении во времени состояний нейтронных каналов представляются в виде диаграммы. График мощности и диаграмма состояний расположены таким образом, что шкала времени для них совпадает. Выбор масштаба координатной сетки производится автоматически в зависимости от длительности представленного отрезка цикла.

Состояние каждого нейтронного канала на диаграмме показывается определенным цветом. На странице представления данных присутствует легенда, поясняющая соответствие цвета на диаграмме состоянию канала. Пример графика и диаграммы представлен на иллюстрации 4.2 в главе 4.

Помимо графической информации, страница представления данных содержит числовые значения следующих рабочих параметров для представленного отрезка времени работы источника:

- Full time – длительность представленного отрезка
- Power-on time – время работы источника

- Idle time – время простоя источника
- Shutdowns – количество отключений источника
- Burn-up – выделенная источником мощность
- Last power – последнее значение мощности в отрезке

Помимо этих параметров на странице представлены значения *времени экспозиции* и *эффективности* для каждого нейтронного канала.

Под *временем экспозиции* понимается суммарный период времени, которое нейтронный канал находился в открытом состоянии при включенном источнике.

Под *эффективностью* понимается отношение времени экспозиции ко времени работы источника, выраженное в процентах.

Пример страницы представления данных расположен на иллюстрации 4.4 в главе 4. Пример индексной страницы расположен на иллюстрации 4.2 в главе 4.

Более подробно работа и устройство приложения мониторинга и представления основных параметров импульсного источника нейтронов рассматривается в главе 3. Руководство пользователя находится в приложении 1. Руководство по установке и настройке программы находится в приложении 2.

## Глава 3: Приложение мониторинга и представления данных

В данной главе подробно рассматривается приложение мониторинга и представления данных.

### 3.1 Выбор языка программирования для реализации

При выборе языка программирования, который был использован для написания приложения, рассматривались следующие критерии:

**Тип аппаратной платформы.** Приложение должно выполняться на сервере лаборатории нейтронной физики ОИЯИ. В настоящее время в качестве сервера ЛНФ используется компьютер Sun под управлением операционной системы Solaris 2.7, следовательно, приложение должно работать на данной аппаратно-программной платформе. Кроме того, принимая во внимание высокую производительность новых моделей компьютеров, основанных на микропроцессорах фирмы Intel, а так же их относительно невысокую стоимость, необходима возможность использования приложения на таких системах, поскольку вероятно их использование в будущем. При этом желательна возможность легкого переноса приложения с одной платформы на другую.

**Работа с физическими данными.** В задачу приложения входит обработка физических данных, таких как значение мощности источника, время работы и т.д. Работа с такими данными должна быть выполнена средствами языка программирования с достаточной точностью.

**Роль серверного приложения.** Программа выполняет задачи, характерные для серверного приложения, такие как взаимодействие с web-сервером, формирование HTML кода, и т.д. Необходимо, чтобы реализация таких функций была предусмотрена создателями языка программирования.

Перечисленным выше требованиям отвечает язык программирования Java. Java является объектно-ориентированным языком программирования, разработанным фирмой Sun Microsystems.

Основные достоинства языка Java:

- Наибольшая степень переносимости программ
- Мощные стандартные библиотеки
- Встроенная поддержка работы в сетях

Основные недостатки языка Java:

- Низкое, в сравнении с другими языками, быстродействие
- Повышенные требования к объему оперативной памяти

Благодаря использованию интерпретатора байт-кода приложения, написанные на языке Java, могут выполняться на различных аппаратно-программных платформах, в том числе на системах, построенных на основе микропроцессоров Intel и SPARC. Интерпретатор языка Java заранее встроен в современные операционные системы.

В Java реализованы необходимые типы данных и математические функции, а так же необходимые для визуализации графические функции.

Данный язык удобен для создания серверных приложений благодаря использованию Java сервлетов и технологии JSP. Помимо этого, язык Java обеспечивает удобный механизм взаимодействия с базами данных, что может быть полезно при дальнейшем развитии информационной системы.

### 3.2 Представление данных

Для организации хранения данных в процессе работы используются специально созданные классы<sup>1</sup>. Далее приводится описание реализованных классов с указанием доступных извне методов и переменных.

**Значение мощности. Класс PowerPoint.** Данный класс предназначен для хранения данных, полученных в результате одного измерения средней мощности источника. Класс содержит следующие переменные:

date – дата и время измерения мощности (java.util.Date)

---

<sup>1</sup> Понятие класса используется в контексте объектно-ориентированного программирования.

`power` – значение мощности источника (`integer`)

`error` – значение относительной погрешности (`double`)

Информация, которую содержит экземпляр класса, соответствует строке файла данных о мощности источника.

**Набор значений мощности. Класс `PowerSet`.** Назначение класса - сохранение ряда значений мощности и предоставление удобного механизма для последовательного обращения к ним. Помимо этого, класс предусматривает сохранение некоторых характеристик набора, таких как максимальное значение мощности и др. Содержит следующие методы:

`put (PowerPoint p)` – добавление значения мощности

`PowerPoint get ()` – считывание очередного элемента набора

`int size ()` – предоставление количества значений в наборе

`reset ()` – обнуление номера последнего считанного значения

`PowerPoint getFirstPoint ()` – считывание первого элемента

`PowerPoint getLastPoint ()` – считывание последнего элемента

Кроме того, класс `PowerSet` содержит следующие переменные:

`maxpower` – максимальное значение мощности за цикл (`integer`)

`full` – длительность цикла в миллисекундах (`long`)

`shutdown` – количество отключений источника за цикл (`integer`)

`poweron` – время работы источника в миллисекундах (`long`)

Информация, которую содержит экземпляр класса, соответствует файлу данных о мощности источника за цикл.

**Состояние нейтронных каналов. Класс `BeamState`.** Данный класс предназначен для хранения данных о состоянии нейтронных каналов, полученных в результате одной регистрации. Класс содержит следующие переменные:

`date` – дата и время регистрация состояния (`java.util.Date`)

`beam` – массив состояний всех каналов (`byte[12]`)

Информация, которую содержит экземпляр класса, соответствует строке файла данных о состоянии каналов.

**Набор состояний каналов. Класс BeamSet.** Назначение класса - сохранение ряда состояний каналов и предоставление удобного механизма для последовательного обращения к ним. Содержит следующие методы:

`put (BeamState p)` – добавление значения мощности  
`BeamState get()` – считывание очередного элемента набора  
`int size()` – предоставление количества значений в наборе  
`reset()` – обнуление номера последнего считанного значения

Информация, которую содержит экземпляр класса, соответствует файлу данных о состоянии нейтронных каналов за цикл.

### 3.3 Компоненты приложения

Функции приложения распределены между отдельными компонентами. Это позволяет упростить программирование и написать удобный для отладки исходный код. Далее подробно описано назначение каждого компонента.

**Компонент загрузки данных. Класс DataSource.** Данный компонент предназначен для загрузки необходимой информации из файлов данных и формирования наборов значений мощности (экземпляр класса `PowerSet`) и состояний нейтронных каналов (экземпляр класса `BeamSet`). Помимо этого, класс формирует список идентификаторов циклов, данные о которых доступны для представления. Класс `DataSource` содержит следующие доступные извне методы:

`PowerSet getPowerSet(String name)` Метод предназначен для формирования набора значений мощности для цикла, имеющего идентификатор, указываемый в параметре метода. В процессе выполнения из файла конфигурации считывается путь к рабочей директории, а из идентификатора цикла формируется имя файла. Затем производится построчное чтение файла данных о мощности источника. При этом из данных, содержащихся в каждой строке, формируется экземпляр класса `PowerPoint`, который добавляется в набор значений мощности (экземпляр класса `PowerSet`). После достижения конца файла метод возвращает заполненный набор.

`BeamSet getBeamSet(String name)` Метод предназначен для формирования набора состояний нейтронных каналов для цикла, имеющего идентификатор, указываемый в параметре метода. Работает аналогично предыдущему методу, за тем исключением, что из каждой строки считанного файла формируется экземпляр класса `BeamState`, который добавляется в набор значений состояний каналов (экземпляр класса `BeamSet`).

`CycleSet getCycleSet()` Метод используется для получения списка идентификаторов цикла, файлы данных которых находятся в рабочей директории. Для этого формируются список файлов данных о мощности и список файлов данных о состоянии каналов. Метод возвращает экземпляр класса, содержащий идентификаторы тех циклов, для которых доступны оба файла.

**Компонент графического представления. Класс `QuickPlot`.** Компонент предназначен для формирования графика мощности импульсного источника и диаграммы состояний каналов транспортировки нейтронов. Выполняемая компонентом функция зависит от того, какой конструктор применяется для создания экземпляра класса:

`QuickPlot(long min, long max, PowerSet ps, BeamSet bs)` создает экземпляр класса, который формирует изображение, содержащее график мощности и диаграмму состояний нейтронных каналов за период времени начиная с `min` и заканчивая `max`.

`QuickPlot(long min, long max, PowerSet ps)` создает экземпляр класса, который формирует изображение, содержащее только график мощности реактора. График представляет данные за период времени начиная с `min` и заканчивая `max`.

`QuickPlot(long min, long max, BeamSet bs)` создает экземпляр класса, который формирует изображение, содержащее только диаграмму состояний

нейтронных каналов. График представляет данные за период времени начиная с `min` и заканчивая `max`.

В процессе работы компонента в памяти создается графический объект `Image` построенный на основе цветовой модели RGB с 8 битным представлением каждого цвета.

Для вычисления координат элементов графика и диаграммы производится вычисление отношения отрезка времени (`max - min`) к горизонтальному размеру графика мощности и диаграммы состояний нейтронных каналов:

$$s\_x = (max - min) / width$$

где `width` – ширина графика мощности и диаграммы состояний.

Так же вычисляется отношение диапазона мощности импульсного источника к вертикальному размеру графика:

$$s\_y = power\_max / power.height$$

где `power_max` – максимальное значение мощности,  
`power.height` – высота графика мощности.

Производится выбор необходимого числа временных меток, необходимых для построения координатной сетки, в зависимости от величины временного диапазона; составление набора, содержащего информацию о позиции каждой метки и текст, обозначающий соответствующие метке дату и время.

Затем выполняется построение границ графика мощности, границ диаграммы состояний нейтронных каналов; построение осей координатной сетки;

При построении графика мощности координаты точек вычисляются по формуле:

$$X = \text{power.x} + (\text{point.date} - \text{min}) / \text{s\_x}$$

$$Y = \text{power.y} + \text{power.height} - \text{point.power} / \text{s\_y}$$

где `power.x` – отступ границы графика от левого края изображения,  
`point.date` – момент времени, соответствующий точке,  
`power.y` – отступ границы графика от верхнего края изображения,  
`power.height` – высота графика мощности,  
`point.power` – значение мощности источника

Все действия по созданию изображения производятся во время выполнения метода `getImage(int width, int height)`, который возвращает графический объект `Image`. В параметрах метода указываются размеры изображения по горизонтали и вертикали соответственно. Это делает компонент довольно универсальным, однако следует заметить, что в том случае, если размер изображения задан относительно небольшим, возможно наложение меток координатной сетки и прочие неприятности. Фактически, на странице приложения мониторинга и представления параметров, используется изображение размером 500 x 500 пикселей.

**Компонент мониторинга и представления. Класс `DreadBean`.** Основной компонент приложения. В процессе выполнения производится загрузка данных с помощью компонента `DataSource`, построение графика мощности и диаграммы состояний нейтронных каналов с помощью компонента `QuickPlot`. Затем выполняется расчет рабочих параметров, таких как:

- количество отключений реактора
- время работы реактора
- энергия, выделенная реактором
- время простоя реактора
- время экспозиции<sup>2</sup> нейтронных каналов
- эффективность работы нейтронных каналов

---

<sup>2</sup> Под *временем экспозиции* понимается суммарный период времени, которое нейтронный канал находился в открытом состоянии при включенном источнике.

Энергия выделенная реактором считается равной интегралу от мощности по времени. Фактически для каждого измерения мощности рассчитывается произведение значения мощности в мегаваттах на отрезок времени до следующего измерения, выраженный в часах. Сумма произведений для представленного отрезка времени равняется выделенной энергии в мегаватт-часах.

В дальнейшем значения полученных параметров включаются в страницу представления данных `status.jsp`.

**Компонент формирования списка циклов. Класс `IndexBean`.** В процессе выполнения компонента производится загрузка набора циклов, данные о которых доступны для представления, и формирование элементов списка циклов. В дальнейшем список используется в индексной странице `index.jsp`.

**Страница представления данных. Файл `status.jsp`.** Компонент является страницей JSP и содержит HTML разметку и инструкции включения значений параметров из компонента `DreadBean`. Страница содержит ссылку на компонент формирования изображения.

**Страница списка доступных циклов. Файл `index.jsp`.** Компонент является страницей JSP и содержит HTML разметку и инструкции включения элементов списка циклов из компонента `IndexBean`.

**Компонент формирования изображения. `image servlet`.** Компонент предназначен для преобразования объекта `Image` в графическое изображение формата GIF. Для формирования изображения используется класс `GIFOutputStream` написанный Кэрри Шетлайном и распространяемый бесплатно.

### 3.4 Рабочий цикл программы

Функции приложения, связанные с формированием страниц в ответ на запросы пользователя, можно условно разделить на две категории:

- формирование индексной страницы
- формирование страницы представления данных

Далее изложена последовательность действий, выполняемая при обработке запросов вышеуказанных страниц.

**Формирование индексной страницы.** При формировании индексной страницы параметры запроса не используются. В случае если в запросе присутствуют параметры, они игнорируются. Выполняется следующая последовательность действий:

1. Считывается список файлов данных в рабочей директории. Создается набор идентификаторов циклов – экземпляр класса CycleSet.
2. На основе данных, содержащихся в наборе идентификаторов, формируются элементы таблицы.
3. Элементы таблицы включаются в JSP страницу index.jsp. Полученная HTML страница передается web-серверу для представления пользователю системы.

Передача сформированной страницы пользователю выполняется web-сервером без участия приложения мониторинга и представления данных.

**Формирование страницы представления данных.** При формировании страницы представления данных приложение получает и анализирует параметры запроса пользователя, если они присутствуют. Параметры указывают информацию о цикле и запрашиваемом временном отрезке в пределах этого цикла.

Используется стандартный механизм передачи параметров, используемый в работе протокола HTTP [5]. Запрос может содержать следующие параметры:

- `cycle` – идентификатор цикла
- `from` – начало временного отрезка
- `to` – конец временного отрезка

В случае отсутствия в запросе какого-либо из перечисленных параметров, для отсутствующего запроса принимаются следующие значения по умолчанию:

- для параметра `cycle` – текущий цикл
- для параметра `from` – начало цикла
- для параметра `to` – конец цикла

Таким образом, запрос без параметров подразумевает запрос полной информации о текущем цикле.

Значение идентификатора цикла представляет собой восьмизначное целое число, которое формируется из даты первого измерения, входящего в цикл и позволяет определить названия файлов данных о мощности и состоянии нейтронных каналов.

В процессе формирования страницы представления данных выполняются следующие действия:

1. Из идентификатора цикла формируются имена файлов данных о мощности и файлов данных о состоянии нейтронных каналов.
2. Файлы данных соответствующего цикла загружаются с сервера. Формируются экземпляры классов `PowerSet` и `BeamSet`.
3. Выполняется построение графика мощности источника и диаграммы состояний нейтронных каналов. Создается графический объект `Image`.
4. Вычисляются основные рабочие параметры источника, такие как время работы, время простоя, выделенная энергия и т. д.
5. Полученные значения рабочих параметров включаются в JSP страницу `status.jsp`. Полученный код HTML-страницы передается web-серверу.
6. Из графического объекта `Image` формируется графический файл в формате GIF. Полученный файл передается web-серверу.

Передача сформированной HTML-страницы и графического файла пользователю выполняется web-сервером без участия приложения.

Иллюстрация 3.4.1 содержит блок-схему, на которой представлено взаимодействие компонентов приложения между собой.

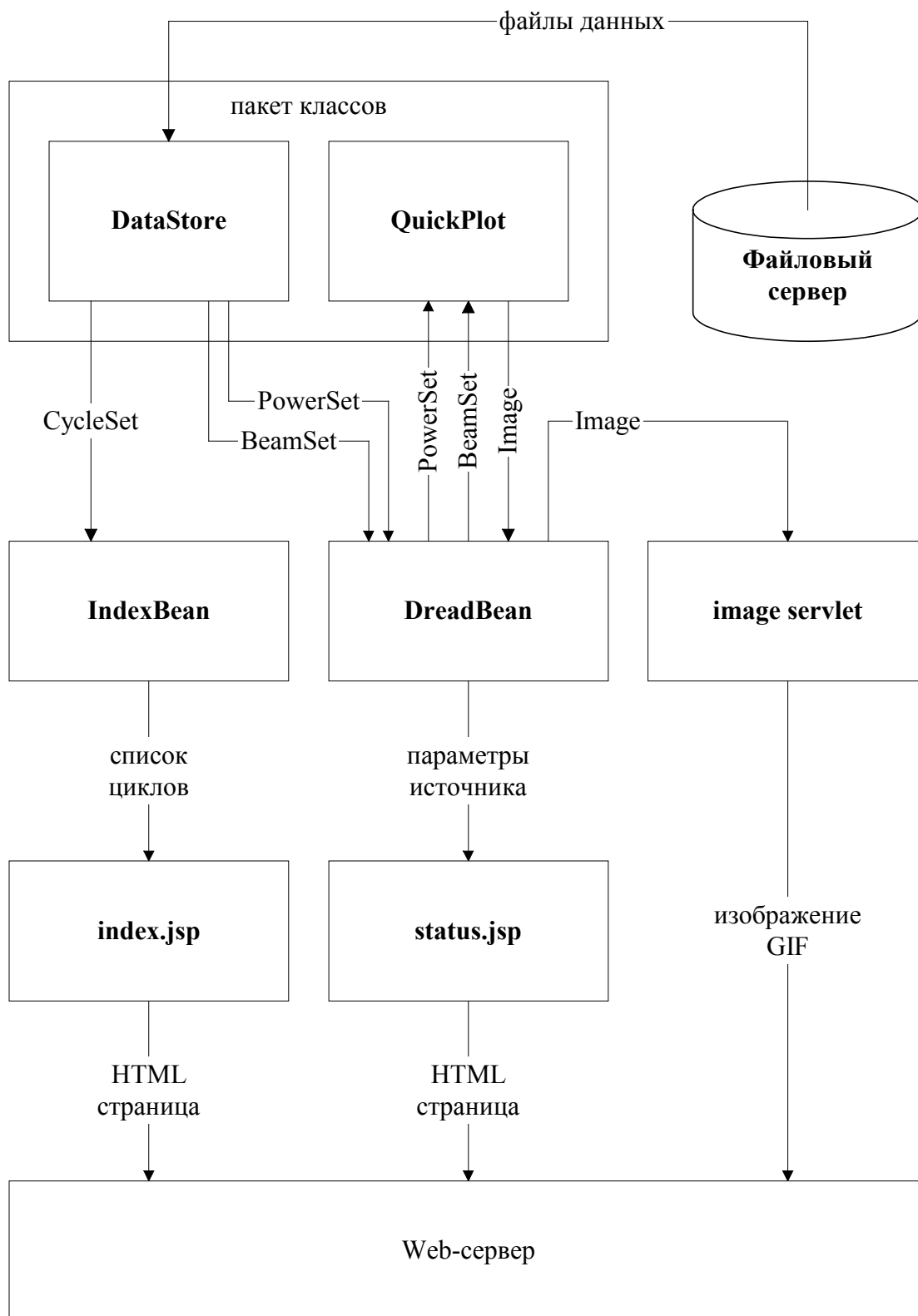


Рис. 3.4.1 Схема взаимодействия компонентов приложения

На иллюстрации 3.4.2 представлена схема преобразования данных в процессе формирования страницы представления рабочих параметров импульсного источника нейтронов.

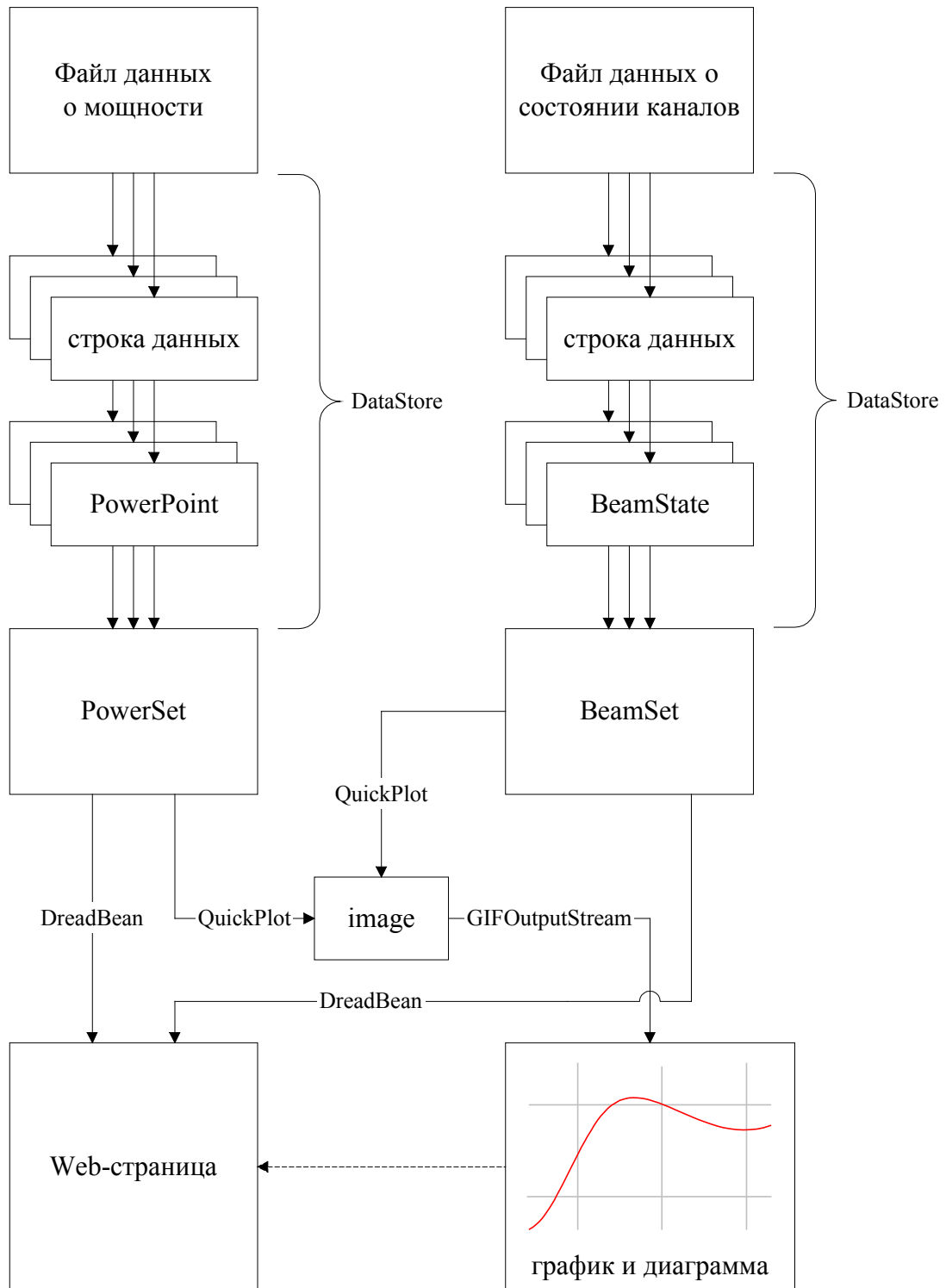


Рис. 3.4.2 Схема преобразования данных

### 3.5 Файл конфигурации программы

Файл конфигурации содержит информацию, необходимую для чтения данных из файлов результатов измерений. Файл имеет название `source.properties` и содержит следующие параметры:

```
source.encoding – кодировка символов файлов данных
source.directory – название директории, содержащей файлы данных
source.timezone – временная зона местоположения источника
```

Каждый параметр задается с начала строки. Значение параметра отделяется от имени символом равенства. Например, строка, содержащая значение параметра кодировки равное `iso8859-1` имеет вид:

```
source.encoding = iso8859-1
```

Подробно о допустимых значениях параметров указано в руководстве по установке и настройке программы.

### 3.6 Обработка ошибок

Под *обработкой ошибок* понимается выполнение приложением некоторых действий в случае возникновения нештатной ситуации, такой как, например, неправильные значения параметров запроса, поврежденный файл данных и т. д. В зависимости от типа ошибки целью обработки данной ошибки является либо сообщение пользователю о возникновении ошибки и ее вероятных причинах, либо обеспечение дальнейшего нормального выполнения приложения.

Для решения таких задач используется механизм обработки исключительных ситуаций, встроенный в язык программирования Java [2]. Он позволяет применить одинаковый подход к обработке как ошибок, возникающих в процессе работы встроенных в Java методов, так и логических ошибок работы приложения.

В программе используются следующие классы исключений:

**BrowseException** Исключение вызывается при возникновении ошибок во время формирования списка доступных для представления циклов. Вызов исключения приводит к возвращению пользователю сообщения, в котором указана причина ошибки.

**LoadException** Исключение вызывается при возникновении ошибок во время загрузки файла данных в память. Возможные причины следующие:

- поврежден файл данных
- отсутствует файл данных
- неверный идентификатор цикла
- неизвестная кодировка символов
- ошибка доступа к файлу

Вызов исключения приводит к возвращению пользователю сообщения, в котором указана причина ошибки.

**ResourceException** Исключение вызывается при возникновении ошибок во время считывания параметров из файла конфигурации. Ниже перечислены следующие возможные причины возникновения ошибки:

- отсутствует файл конфигурации
- отсутствует необходимый параметр

Вызов исключения приводит к возвращению пользователю сообщения, в котором указана причина ошибки. В случае отсутствия необходимого параметра пользователь получает сообщение об ошибке с указанием параметра.

**PaintException** Исключение вызывается при возникновении ошибок во время построения графика мощности источника и диаграммы состояний нейтронных каналов.

В случае если нормальное завершение приложения невозможно, пользователь получает сообщение об ошибке вместо данных мониторинга. Пример страницы сообщения об ошибке представлен на иллюстрации 4.4 в главе 4.

## Глава 4. Результаты опытной эксплуатации приложения

После написания приложения мониторинга и представления рабочих параметров импульсного источника нейтронов ИБР-2 была проведена его опытная эксплуатация в составе информационной системы. В ходе эксплуатации были выявлены и устранены некоторые ошибки и недостатки программы. На иллюстрации 4.1 приведен пример индексной страницы, формируемой приложением. На иллюстрации 4.2 представлен пример графика мощности и диаграммы состояний нейтронных каналов.

### IBR-2 Monitoring

Distributed Information System for IBR-2 pulsed neutron source users

#### Current Cycle

Cycle ID	Start Date	Finish Date
<a href="#">10022003</a>	February 10, 2003	February 17, 2003

The status of current cycle is **active**.

#### Archived Cycles

Cycle ID	Start Date	Finish Date
<a href="#">13012003</a>	January 14, 2003	January 25, 2003
<a href="#">02122002</a>	December 2, 2002	December 13, 2002
<a href="#">11112002</a>	November 11, 2002	November 22, 2002
<a href="#">21102002</a>	October 21, 2002	November 1, 2002
<a href="#">13052002</a>	May 13, 2002	May 25, 2002
<a href="#">08042002</a>	April 8, 2002	April 19, 2002
<a href="#">11032002</a>	March 11, 2002	March 22, 2002
<a href="#">11022002</a>	February 11, 2002	February 22, 2002
<a href="#">14012002</a>	January 14, 2002	January 26, 2002
<a href="#">12112001</a>	November 12, 2001	November 23, 2001
<a href="#">22102001</a>	October 22, 2001	October 22, 2001

#### Extra Info

[User's manual](#) - Instruction manual for information system users  
[Plans](#) of the IBR-2 operation  
[IBR-2](#) - Information on IBR-2 pulsed neutron source  
[FLNP home](#) - Frank Laboratory for Neutron Physics website

Рис. 4.1 Пример индексной страницы

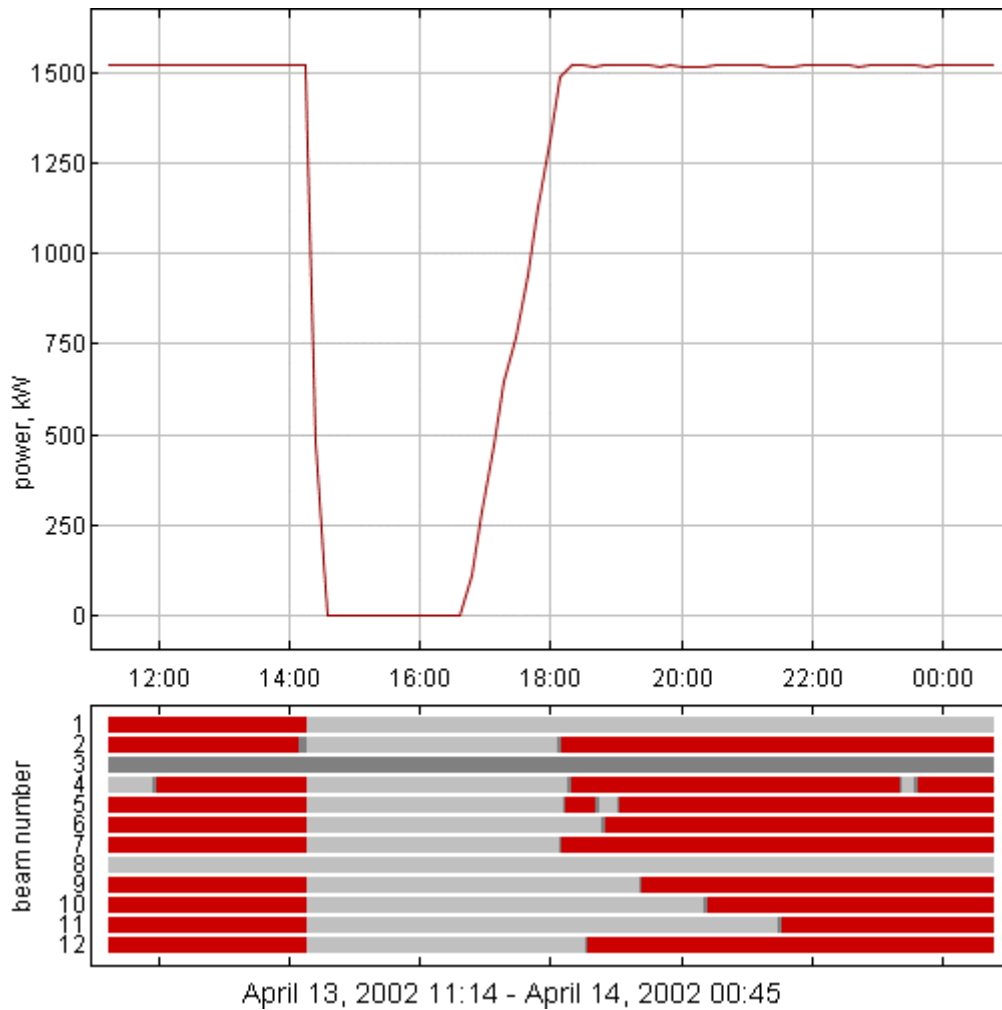


Рис 4.2 Пример графика мощности источника и диаграммы состояния нейтронных каналов

На иллюстрации 4.3 приведен пример страницы, содержащей сообщение об ошибке.

**IBR-2 Monitoring**

Distributed Information System for IBR-2 pulsed neutron source users

**Server error**

*Error message: unable to load configuration: "source.properties" is missing*

Please refer to the [user's manual](#) for instructions.

Рис. 4.3 Пример страницы сообщения об ошибке

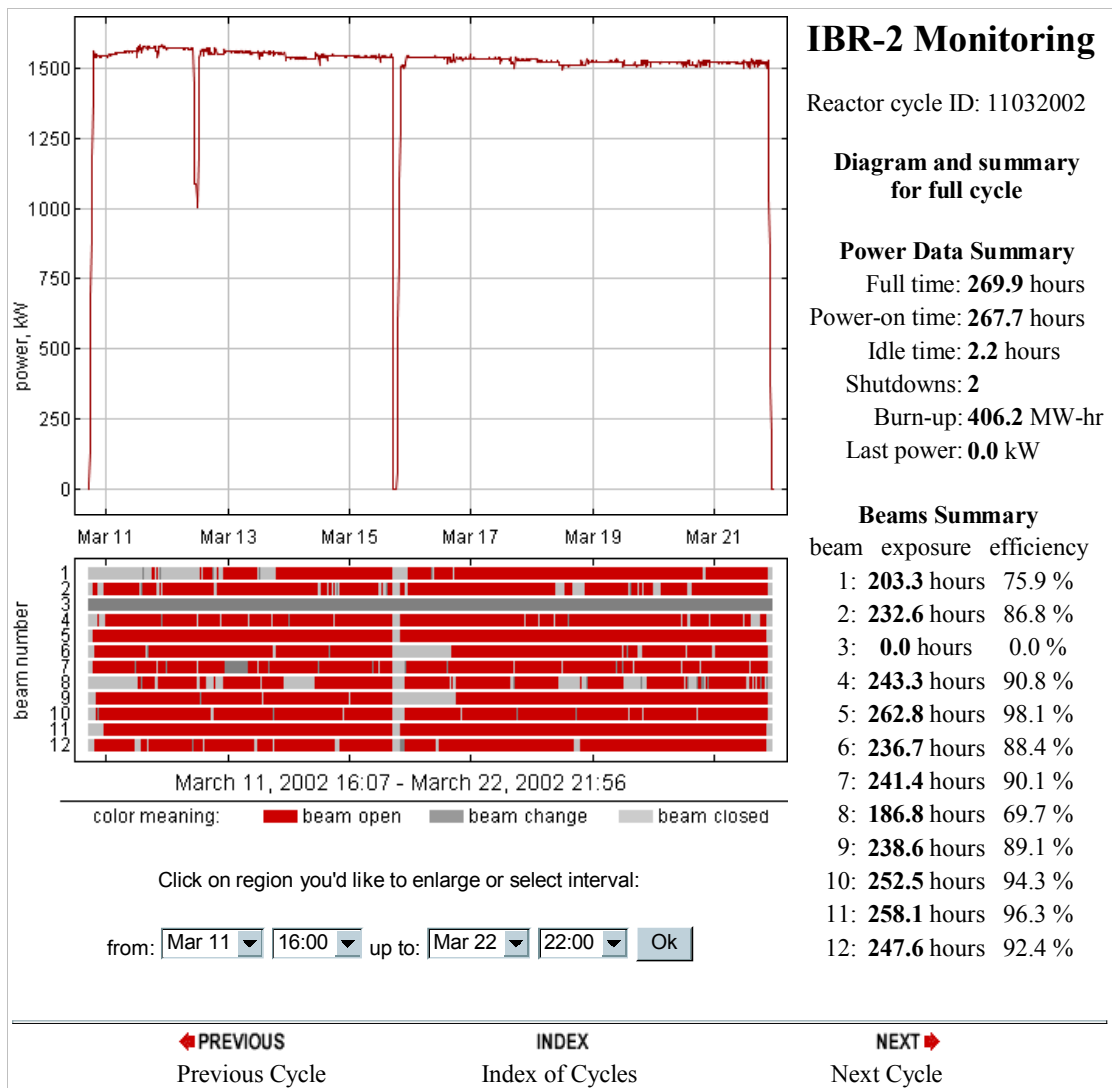


Рис. 4.4 Пример страницы представления параметров

На иллюстрации 4.4 приведен пример страницы представления параметров, формируемой приложением.

По окончании опытной эксплуатации было составлено руководство пользователя информационной системы и руководство по установке и настройке приложения мониторинга и представления параметров источника.

## Заключение

В процессе выполнения дипломного проекта были разработаны структура и алгоритм работы приложения мониторинга и представления рабочих параметров импульсного источника нейтронов. При этом были учтены особенности реализации других компонентов информационной системы.

Результатом работы стало создание программного обеспечения, выполняемого на сервере, обслуживающем пользователей информационной системы. Серверное программное обеспечение написано на языке Java с использованием среды разработки Java Development Kit 1.4.

Была проведена опытная эксплуатация программного обеспечения мониторинга и представления в составе информационной системы. По результатам тестирования были устранены некоторые недостатки. В частности были обнаружены и исправлены неточности в реализации алгоритма автоматического выбора масштаба координатной сетки. После этого были написаны руководство пользователя информационной системы и руководство по установке и настройке приложения мониторинга и представления.

## Литература

1. *B. H. McCormick, T. A. DeFanti, M. D. Brown*, Visualization in Scientific Computing, Computer Graphics Vol. 21, No. 6, November 1987
2. *S. Pelouch*, Java Language Specification, Sun Microsystems, 2000
3. *Б. Кеннеди, Ч. Муссиано*, HTML и XHTML. Подробное руководство, Символ-плюс 2002
4. *П. Л. Карабин*, Язык программирования Java, Познавательная книга 2001
5. *Б. Кришнамурти, Дж. Рексфорд*, Web-протоколы. Теория и практика, Бинум 2002
6. *Г. Корн, Т. Корн* Справочник по математике, Москва 1977
7. *Кассандрова О.Н., Лебедев В.В.* Обработка результатов наблюдений, Наука 1970

## Приложение 1: Руководство пользователя информационной системы

### 1.1 Необходимое программное обеспечение

Для того чтобы получить доступ к информационной системе используйте web-browser (Microsoft Internet Explorer или Netscape Navigator).

### 1.2 Выбор цикла для представления

Для выбора цикла, информацию о котором вам необходимо получить, используйте *индексную страницу*, которая расположена по адресу:

```
http://nfpin.jinr.ru/seamus/
```

Данная страница содержит список циклов, информация о которых доступна для представления. Каждая запись в списке имеет идентификатор цикла, дату начала цикла и дату окончания цикла (последнего измерения). Список отсортирован таким образом, что самый последний цикл расположен на первой строке. Кроме того, на странице указано состояние текущего цикла.

Строка, содержащая идентификатор цикла, является ссылкой на страницу представления данных и содержит номер цикла в параметре.

### 1.3 Работа с программой представления

Страница представления данных расположена по следующему адресу:

```
http://nfpin.jinr.ru/seamus/status.jsp
```

Если использовать приведенный адрес без изменений, то будет выведена информация о текущем (последнем) цикле реактора. Для получения информации о предыдущих циклах вы можете указать идентификатор цикла в качестве параметра запроса:

```
http://nfpin.jinr.ru/seamus/status.jsp?cycle=<cycle_id>
```

где `<cycle_id>` - идентификатор цикла, информация о котором вас интересует. Следует заметить, что указывать идентификатор цикла вручную в параметре запроса не удобно и чревато возможными ошибками. Для выбора цикла рекомендуется использовать индексную страницу.

Программа представления данных формирует страницу, содержащую информацию либо о полном цикле, либо о его части. На странице представлен график мощности реактора, диаграмма состояний нейтронных каналов и основные рабочие параметры за требуемый период.

Для того чтобы получить более детальную информацию по какому-либо временному отрезку, вы можете воспользоваться формой, в которой задаются начало и конец интересующего вас временного отрезка. Кроме того, вы можете получить детальную информацию, кликнув по выбранному участку графика мощности или диаграммы состояний.

Для возвращения в режим представления полного цикла вы можете воспользоваться ссылкой **full cycle**, которая расположена справа от графика.

#### **1.4 Ошибки при работе системы**

В случае возникновения ошибок при работе системы, формируется соответствующее сообщение. Далее рассматриваются возможные сообщения об ошибках и причины их возникновения.

**line N not properly understood** Ошибка при чтении из файла данных строки с указанным номером. Вероятно, поврежден файл данных. Сообщите администратору системы идентификатор цикла, номер строки и дату возникновения ошибки.

**file not found** Ошибка при попытке чтения файла данных. Не найден необходимый файл данных для указанного цикла. Наиболее вероятная причина ошибки – неправильно указанный идентификатор цикла. Воспользуйтесь ссылкой на индексной странице, для доступа к информации об интересующем вас цикле.

**file read error** Ошибка при попытке чтения файла данных. Сообщите администратору системы идентификатор цикла и дату возникновения ошибки.

**source directory is not exists or not a directory** Ошибка при попытке получить список доступных для представления циклов. Наиболее вероятная причина возникновения ошибки – неправильный путь рабочей директории в файле конфигурации программы.

**source directory access denied** Ошибка при попытке получить список доступных для представления циклов. Причина возникновения – неправильно установленные права доступа к рабочей директории.

**file “name” not properly understood** Ошибка при попытке получить из файла данных дату начала цикла и дату окончания цикла. Вероятно, поврежден файл данных. Сообщите администратору системы идентификатор цикла и дату возникновения ошибки.

**configuration file “source.properties” is missing** Ошибка при попытке получить значения конфигурационных параметров приложения. Отсутствует файл конфигурации.

**missing “name” parameter** Ошибка при попытке получить значение конфигурационного параметра *name*. Причина – отсутствие строки, соответствующей указанному параметру в файле конфигурации. О том, какие параметры должен содержать файл конфигурации, указано в руководстве по установке и настройке приложения.

## Приложение 2: Руководство по установке и настройке программы

Программное обеспечение мониторинга и представления рабочих параметров импульсного источника нейтронов устанавливается на сервере, предназначенном для обслуживания пользователей системы.

### 2.1 Необходимое программное обеспечение

Для работы приложения мониторинга и представления рабочих параметров импульсного источника нейтронов убедитесь в том, что в системе установлены следующие программные пакеты:

- интерпретатор языка Java
- контейнер серверных приложений Java
- web-сервер

Ниже даны некоторые комментарии по каждому из перечисленных пакетов.

**Интерпретатор языка Java.** Приложение разработано с учетом совместимости с интерпретатором Java версии 1.1, поэтому вы можете использовать любую версию, начиная с 1.1. Однако следует учитывать, что реализация графических функций в ранних версиях Java, имеет особенности, не позволяющие использовать эти функции, когда доступ приложения к дисплею<sup>3</sup> отсутствует. В этом случае необходимо использование Java версии 1.4.

**Контейнер серверных приложений Java.** Вы можете использовать любой контейнер серверных приложений Java, который поддерживает спецификации Servlet API 2.0 и JSP 1.1. Фактически, если у вас нет никаких специальных требований, вам следует использовать *Jakarta Tomcat web-server*.

**Web-сервер.** Если вы уже используете web-сервер, вам необходимо обеспечить обработку запросов к серверным приложениям с помощью контейнера, такого

---

<sup>3</sup> В данном случае термин дисплей используется в контексте X Window System.

как Jakarta Tomcat. Если web-сервер не установлен, то вы можете возложить функции web-сервера на контейнер серверных приложений Java.

## 2.2 Установка программного обеспечения

Для установки программного обеспечения необходимо поместить файлы, содержащиеся в архиве, в директорию приложений контейнера. Для контейнера Jakarta Tomcat данная директория имеет следующий путь:

```
$TOMCAT_HOME/webapps
```

Для распаковки архива воспользуйтесь командой

```
jar xvf <file_name>
```

где <file\_name> - имя архивного файла приложения.

## 2.3 Настройка файла конфигурации

После установки приложения файл конфигурации будет иметь следующий путь:

```
$TOMCAT_HOME/webapps/seamus/WEB-INF/classes/source.properties
```

Файл конфигурации содержит значения параметров, необходимых для работы приложения. Название параметра отделяется от его значения знаком равенства.

Ниже даны названия параметров, которые должны содержаться в файле конфигурации, и значения этих параметров.

**source.encoding** – кодировка символов, используемая в файле данных. На данный момент используется стандартная кодировка ISO8859-1.

**source.directory** – путь к рабочей директории, содержащей файлы данных. Обратите внимание: для корректной работы приложения необходимо наличие символа косой черты в конце пути.

**source.timezone** – временная зона, в которой находится измерительная аппаратура. Для города Дубны, где расположен Объединенный Институт Ядерных Исследований корректное значение GMT+3.