

**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное учреждение науки**

**«Институт ядерной физики им. Г.И. Будкера СО РАН»**

УТВЕРЖДАЮ

директор ИЯФ СО РАН,

академик \_\_\_\_\_ П.В.Логачев

«\_\_\_\_\_» \_\_\_\_\_ 2018 г.

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ФИЗИЧЕСКОМ ЭКСПЕРИМЕНТЕ.**

Рабочая программа дисциплины

Направление подготовки

**03.06.01 Физика и астрономия (уровень подготовки кадров высшей квалификации)**

Общая трудоемкость дисциплины: 3 зачетные единицы – **108** часа.

Форма промежуточной аттестации: зачет, кандидатский экзамен

Виды деятельности:

Лекции	36	контактная работа обучающихся с преподавателем	76
Семинарские занятия	36		
Самостоятельная работа	28	занятия в активной и интерактивной форме	
Консультации	4		
Зачеты	4	Экзамены	

Рабочая программа дисциплины «Компьютерные технологии в физическом эксперименте», предназначенная для аспирантов ИЯФ СО РАН, разработана в 2018 году в соответствии с ФГОС ВО по направлению подготовки 03.06.01 Физика и астрономия (уровень подготовки кадров высшей квалификации)

Составили:

к.ф.-м.н. Логашенко И.Б.

Рабочая программа

## Содержание

Аннотация.....	4
1. Цели освоения дисциплины.....	5
2. Место дисциплины в структуре ООП.....	5
3. Компетенции обучающегося, формируемые при освоении дисциплины.....	5
4. Структура и содержание дисциплины.....	6
5. Образовательные технологии.....	10
6. Учебно-методическое обеспечение самостоятельной работы студентов.....	11
7. Фонд оценочных средств для проведения аттестации по итогам освоения дисциплины: показатели, критерии оценивания компетенций, типовые контрольные задания.....	11
8. Учебно-методическое и информационное обеспечение дисциплины Рекомендованная литература к теоретическому курсу.....	11
9. Материально-техническое обеспечение дисциплины.....	12

## Аннотация

Программа курса «Компьютерные технологии в физическом эксперименте» составлена в соответствии с требованиями федерального государственного стандарта высшего образования по направлению подготовки 03.06.01 Физика и астрономия (уровень подготовки кадров высшей квалификации).

Дисциплина входит в набор элективных дисциплин (дисциплин по выбору аспиранта), предлагаемых аспирантам всех профилей подготовки, обучающимся в аспирантуре ИЯФ СО РАН.

Цель курса – ознакомление аспирантов с большим спектром информационных технологий, ставших неотъемлемой частью любого физического эксперимента.

Дисциплина нацелена на формирование у выпускника следующих компетенций:

- готовность участвовать в работе российских и международных исследовательских коллективов по решению научных и научно-образовательных задач (УК-3);
- готовность использовать современные методы и технологии научной коммуникации на государственном и иностранном языках (УК-4);
- способность самостоятельно осуществлять научно-исследовательскую деятельность в соответствующей профессиональной области с использованием современных методов исследования и информационно-коммуникационных технологий (ОПК-1).

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, консультации, самостоятельная работа студента, зачет.

Программой дисциплины предусмотрены следующие виды контроля:

Текущий контроль: посещаемость, доклад по выбранной теме.

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов, в том числе 36 часов лекций, 36 часов практических занятий, 36 часа занятий в интерактивной форме.

## **1. Цели освоения дисциплины**

Целью освоения дисциплины «Компьютерные технологии в физическом эксперименте» является ознакомление аспирантов с большим спектром информационных технологий, ставших неотъемлемой частью любого физического эксперимента.

Современные физические исследования на сегодняшний день зачастую требуют широкого использования информационных технологий. Проведение эксперимента часто сопряжено с набором огромного количества данных, от десятков терабайт в небольших экспериментах до сотен петабайт в самых крупных экспериментах. Организация сбора, хранения и обработки такого объема данных – сложная задача, требующая использования самых передовых достижений в области информационных технологий. В связи с уникальностью каждого эксперимента для анализа набранных данных требуется разрабатывать специализированное программное обеспечение. Теоретические расчеты, особенно в области физики элементарных частиц, требуют огромных вычислительных ресурсов, и для них используются ресурсы суперкомпьютерных центров. Кроме того, современные исследования часто ведутся большими коллективами ученых. Все перечисленное приводит к тому, что физик должен ориентироваться в широком круге вопросов, связанных с информационными технологиями, и владеть основными приемами совместной работы и совместной разработки программного обеспечения.

## **2. Место дисциплины в структуре магистерской программы**

Дисциплина «Компьютерные технологии в физическом эксперименте» относится к вариативной части блока 1 «Дисциплины (модули)».

Студенты, приступающие к изучению этой дисциплины, должны иметь общую базовую подготовку в рамках программы 5-6 лет обучения в ВУЗе, в том числе:

- иметь общее представление об организации физического эксперимента,
- знать основы объектно-ориентированного программирования.

## **3. Компетенции обучающегося, формируемые в результате освоения дисциплины**

Дисциплина «Компьютерные технологии в физическом эксперименте» нацелена на формирование у выпускника следующих компетенций:

- готовность участвовать в работе российских и международных исследовательских коллективов по решению научных и научно-образовательных задач (УК-3);
- готовность использовать современные методы и технологии научной коммуникации на государственном и иностранном языках (УК-4);
- способность самостоятельно осуществлять научно-исследовательскую деятельность в соответствующей профессиональной области с использованием современных методов исследования и информационно-коммуникационных технологий (ОПК-1);

В результате освоения дисциплины обучающийся должен:

- Знать принципы организации систем сбора, обработки, моделирования и хранения данных в современном физическом эксперименте, основы технологии реляционных баз данных, основные приемы проектирования программных комплексов.
- Уметь проектировать системы сбора, обработки, моделирования и хранения данных для современного физического эксперимента; организовывать процесс обработки данных эксперимента; совместно работать в команде при создании специализированного программного обеспечения эксперимента; решать типичные задачи, возникающие при анализе данных современного физического эксперимента.

#### 4. Структура и содержание дисциплины

Дисциплина «компьютерные технологии в физическом эксперименте» представляет собой полугодовой курс, читаемый в аспирантуре ИЯФ СО РАН. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов.

№ п/п	Раздел дисциплины	Не- деля се- мест- ра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				
			Все- го	Аудиторные часы		Сам. ра- бота в течение семестра (не включая период сессии)	Промежу- точная ат- тестация (подготов- ка к зачету и зачет)
				Лекции (кол-во часов)	Практи- ческие занятия (кол-во часов)		
1	Введение	1	2	2			
2	Системы обработки данных физического эксперимента.	2-3	10	4	4	2	
3	Система сбора данных современного эксперимента.	4-5	10	4	4	2	
4	Моделирование эксперимента.	6-7	10	4	4	2	
5	Языки программирования и основы проектирования.	8-10	12	6	4	2	
6	Введение в реляционные базы данных.	11-13	12	6	4	2	

7	Дополнительные главы	14-18	30	10	10	10	
8	Доклады слушателей по избранным темам.	14-18	22		6	12	
	Зачет по дисциплине						4
	Итого по курсу:		108	36	36	32	4

### Программа курса лекций

1. Введение. Роль информационных технологий в современном физическом эксперименте. История развития вычислительной техники. Архитектура современного компьютера, архитектуры процессоров. История развития операционных систем. Основные функции операционных систем. Процессы и потоки. Планирование процессов. Управление памятью. Виртуальная память. Сегментирование памяти. Ввод-вывод, файловая система. Файловая система ext2. Журналируемые файловые системы. Распределенные файловые системы.
2. Системы обработки данных физического эксперимента. Основные этапы обработки: предварительная обработка, реконструкция, анализ. Модульная организация программного обеспечения обработки данных, понятие программного каркаса. Программный каркас GAUDI. Роль моделирования в обработке данных. Организация научного анализа данных. Непредвзятый анализ. Систематические ошибки.
3. Система сбора данных современного эксперимента. Регистрация случайных событий и понятие триггера. Роль буферизации в повышении производительности системы сбора данных. Конвейерный принцип организации триггера. Многоуровневый триггер. Система сбора данных большого эксперимента. Шина и сеть как различные топологии подключения оцифровывающей аппаратуры. Этап сборки события. Параметры систем сбора данных известных экспериментов. Системы управления и мониторинга. Организация обмена данными между процессами системы. Понятие конечного автомата.
4. Моделирование эксперимента. Роль моделирования при проектировании, проведении и анализе данных эксперимента в физике высоких энергий. Моделирование эффективности и акцептанса. Существующие программные пакеты для моделирования эксперимента. Программный пакет Geant4. Сфера применения Geant4. Использование метода Монте-Карло для моделирования взаимодействия частиц с веществом. Базовый алгоритм моделирования в случае одного процесса, нескольких процессов. Реализация метода Монте-Карло в Geant4. Обзор процессов взаимодействия, включенных в пакет Geant4. Ядро Geant4, организация взаимодействия конечного пользователя с ядром. Организация процесса моделирования, понятие захода, события, трека, шага моделирования. Программная организация Geant4, список базовых классов, обязательных для реализации. Описание геометрии и вещества детектора. Описание списка возможных процессов взаимодействия. Первичный генератор. Другие интерфейсные классы. Сохранение информации о результатах моделирования.

5. Языки программирования и основы проектирования. Языки программирования. Понятие парадигмы программирования. Декларативная и императивные парадигмы. Классификация и история развития языков программирования. Этапы компиляции программы. Формальные грамматики и их связь с языками программирования. Иерархия Хомского. Утилиты yacc и lex, реализация собственного языка программирования. Парадигма объектно-ориентированного программирования. Основные понятия ООП: инкапсуляция, наследование, полиморфизм. Язык C++. Основные черты C++: классы и объекты, шаблоны, исключения, поддержка обобщенного программирования. Процесс разработки программного обеспечения. Основные модели разработки. Rational Unified Process. Agile Software Development. Язык UML. Основные типы диаграмм в UML. Принципы проектирования. Понятие паттернов программирования. Основные паттерны: Стратегия, Шаблонный Метод, Декоратор, Наблюдатель, Фабричный метод, Абстрактная фабрика, Одиночка, Команда, Адаптер, Фасад, Итератор, Компоновщик, Заместитель. Составной паттерн Model-View-Controller. Примеры использования паттернов в Geant4 и других программных продуктах.
6. Введение в реляционные базы данных. Базы данных. Реляционная модель. Принципы проектирования баз данных. Концептуальная, логическая и физическая модели базы данных. Нормальные формы, нормализация базы данных. Язык SQL. Команды CREATE, INSERT, UPDATE, DELETE, SELECT. Чтение данных из нескольких таблиц. Представления. Агрегатные функции и запросы. Вложенные запросы. Процедурные расширения языка SQL. Транзакции. Хранение неструктурированных данных в базе данных. Язык XML и связанные с ним технологии. Интеграция технологий XML и реляционных баз данных. Хранение бинарных данных в реляционных СУБД. Нереляционные базы данных. Модель MapReduce.
7. Сетевые технологии. Причины и история возникновения компьютерных сетей. Топологии сетей. Глобальные и локальные сети. Протоколы, службы и сервисы. Модель OSI. Модель TCP/IP. Физический уровень. Физическая среда передачи данных. Проверка и коррекция ошибок при передаче, коды Хэмминга. Передача данных в сетях Ethernet, Fast Ethernet, Gigabit Ethernet. Беспроводные сети. Канальный уровень. MAC-адрес, протокол Ethernet. Примеры других протоколов канального уровня. Мосты и коммутаторы. Сетевой уровень. IP-адрес, протокол IP. Протокол ARP. Маршрутизация. Проблема нехватки IP адресов. Преобразование сетевых адресов (NAT). Протокол IPv6. Транспортный уровень. Связь с установлением соединения и без него. Протоколы UDP и TCP. Трансляция имен в адреса, DNS. Качество обслуживания компьютерных сетей. Механизмы управления качеством обслуживания. Прикладной уровень. Модель клиент-сервер. Протокол HTTP. Принципы построения веб-интерфейсов, веб-приложений. CGI, AJAX. Примеры других протоколов прикладного уровня.
8. Компьютерная безопасность. Необходимость шифрования при передаче данных. Фундаментальные принципы криптографии. Принцип Керкгофа. Идеальный шифр – одноразовый блокнот. Квантовая криптография, протокол квантового распределения ключа BB84. Шифрование с симметричным ключом. Алгоритмы DES, 3DES, AES. Шифрование с открытым ключом. Алгоритм RSA. Обмен ключами, алгоритм Диффи-Хелмана. Хэш-



функции, профиль сообщения, функции MD5, SHA. Задача о днях рождения. Цифровая подпись. Понятие сертификатов. Стандарт X.509. Инфраструктура систем с открытыми ключами. Аутентификация с симметричным шифрованием. Аутентификация с помощью центра распространения ключей. Протокол Kerberos.

9. Динамические языки, язык Python. Динамические языки. Прототипирование программных систем с помощью динамических языков. История развития скриптовых и динамических языков. Язык Python и его использование в физике высоких энергий. Основные синтаксические конструкции языка. Основные встроенные типы данных, кортежи, списки, словари. Реализация механизма исключений в Python. Стандартные модули Python. Поддержка объектно-ориентированного программирования. Поддержка функционального программирования. Численные расчеты с использованием Python. Реализация сетевых приложений с использованием Python. Работа с реляционными базами данных. Создание графического интерфейса. Разработка веб-приложений. Интеграция Python с другими языками программирования. Интеграция Python с программными пакетами ROOT и Geant4.
10. ГРИД и облачные технологии. Многоуровневая вычислительная система экспериментов на Большом Адронном Коллайдере. Иерархия вычислительных систем: процессор, сервер, кластер, ГРИД. Вычислительный кластер: операционные системы, сетевые технологии. Организация распределенных файловых систем для вычислительных кластеров. Объединение ресурсов нескольких вычислительных кластеров, архитектура ГРИД. Обзор существующих ГРИД систем. Аутентификация пользователей в ГРИД системах, виртуальные организации. Управление ресурсами в ГРИД системах. Пример использования ГРИД системы для обработки данных эксперимента. Технологии виртуализации. Концепция облачных вычислений. Модели предоставления ресурсов: инфраструктура как сервис (IaaS), программное обеспечение как сервис (SaaS), платформы как сервис (PaaS). Обзор существующих облачных сервисов. Пример одной коммерческой платформы облачных вычислений (например, Amazon Elastic Compute Cloud). Перспективы использования коммерческих облаков для моделирования данных экспериментов. Обзор локальных суперкомпьютерных ресурсов.
11. Параллельное программирование, вычисления на видеокартах. Архитектура параллельных вычислительных систем. Измерение производительности. Пиковая и реальная производительность. Суперскалярные и VLIW-архитектуры. Основные классы современных параллельных компьютеров: параллельные векторные системы, массивно-параллельные системы, симметричные мультипроцессорные системы с общей памятью, системы с неоднородным доступом к памяти, кластерные системы. Особенности языков программирования с поддержкой параллельных и конвейерных вычислений. Параллельные процессы. Взаимодействия и синхронизация параллельных процессов. Взаимное исключение. Двоичные и считающие семафоры. Мониторы. Механизмы передачи сообщений. Рандеву. Технологии параллельного программирования MPI, OpenMP. Основные понятия теории сетей Петри. Графический ускоритель как настольный суперкомпьютер. Архитектура графических ускорителей ATI и NVIDIA. Классы задач, эффективно решаемые с использованием графических ускорителей. Технологии CUDA и OpenCL. Введение в техноло-

гию CUDA, иерархия памяти. Пример использования вычислений на графических ускорителях для обработки данных физического эксперимента, для вычислений на решетках. Другие типы спецвычислителей.

12. Языки описания электроники. Специализированная электроника детекторных систем и триггера в экспериментах в области физики высоких энергий. Программируемая логика (ПЛИС, FPGA). Область применения ПЛИС. Существующие решения Xilinx, Altera. Типичные составляющие элементы ПЛИС. Технологии разработки устройств с применением ПЛИС. Специализированные языки программирования AHDL, VHDL. Пример описания простой электронной схемы. Примеры использования ПЛИС в экспериментах ФВЭ. Использование ПЛИС в качестве спецвычислителя.
13. Введение в функциональное программирование. Парадигмы программирования. Парадигма функционального программирования, ее сравнение с императивным и логическим программированием. Введение в  $\lambda$ -исчисление.  $\lambda$ -исчисление как универсальный язык программирования. Обзор существующих языков функционального программирования. Введение в язык LISP. Поддержка функционального программирования в универсальных языках. Пример функционального программирования на языке Python. Параллельные вычисления с использованием языков функционального программирования.

Разделы 7-13 являются дополнительными. Лектор выбирает из них 3-4 темы исходя из предпочтений слушателей.

## **5. Образовательные технологии**

Теоретический материал курса освещается в ходе лекций. В лекциях обсуждается как теоретические аспекты, так и реальные примеры использования обсуждаемых методов и технологий из практики наиболее известных экспериментов в мировой науке. Все лекционные занятия проводятся в интерактивной форме. В ходе лекций поощряются вопросы слушателей, часть тем обсуждается в форме дискуссий. Материал всех лекций доступен в электронном виде. В ходе лекций широко используются компьютерные демонстрации. Для чтения отдельных лекций (в особенности, по темам из дополнительного раздела курса) рекомендуется приглашать специалистов, занимающихся активной научной и практической деятельностью в соответствующей области знаний.

Практические занятия проводятся в терминальном классе. В ходе практических занятий слушатели знакомятся с различными библиотеками и пакетами программ, обсуждаемыми на лекциях. Часть практических занятий проводится в форме докладов слушателей, в которых они более детально раскрывают отдельные темы или дают обзор определенного программного продукта. Тема доклада может быть связана с научной работой слушателя.

Текущий контроль по дисциплине осуществляется по посещаемости и по результатам представления доклада на выбранную тему. Промежуточная аттестация – зачет по курсу

## **6. Учебно-методическое обеспечение самостоятельной работы студентов.**

Самостоятельная работа студентов поддерживается следующими методическими пособиями:

1. Логашенко И.Б. Компьютерные технологии в ФЭЧ. Электронный лекционный курс / Новосибирск: НГУ, 2012.

Система контроля включает текущий (по ходу семестра) контроль освоения практического материала.

В качестве текущего контроля успеваемости используется контроль посещаемости и результаты представления доклада на выбранную тему.

### **7. Фонд оценочных средств для проведения аттестации по итогам освоения дисциплины: показатели, критерии оценивания компетенций, типовые контрольные задания**

Примеры тем докладов:

1. Тема, связанная с научной работой аспиранта.
2. Нейрокомпьютеры. Использование нейронных сетей для формирования триггера и для поиска редких процессов в экспериментах в области физики элементарных частиц.
3. R – язык программирования и среда для проведения статистического анализа.
4. Введение в решеточные вычисления.
5. Организация системы сбора данных детектора ATLAS (CMS, LHCb, Belle, детекторов ИЯФ).
6. Организация триггера детектора ATLAS (CMS, LHCb, Belle, детекторов ИЯФ).
7. Организация системы обработки и хранения данных детектора ATLAS (CMS, LHCb, Belle, детекторов ИЯФ).
8. Организация анализа данных в большой научной коллаборации (на примере коллаборации ATLAS, CMS, Belle, Babar).
9. Организация больших программных проектов в ФВЭ, используемые технологии.
10. Система хранения данных Castor.
11. Программный каркас Gaudi.
12. Визуализация данных в детекторе элементарных частиц.
13. Управление данными в экспериментах на LHC.
14. Моделирование адронных процессов в Geant4.
15. Параллельная обработка больших объемов данных в пакете ROOT.
16. Система баз данных эксперимента (выбрать конкретный эксперимент).
17. Использование нереляционных баз данных в физических и астрофизических экспериментах.
18. Системы сбора данных географически распределенных экспериментов (на примере изучения космических лучей).
19. Системы сбора данных экспериментов, базирующихся в космосе.

### **8. Учебно-методическое и информационное обеспечение дисциплины**

а) основная литература:

1. Логашенко И.Б. Компьютерные технологии в ФЭЧ. Электронный лекционный курс.

б) дополнительная литература:

1. К. Групен, Детекторы элементарных частиц, Сибирский хронограф, Новосибирск, 1999.
2. *Онучин А. П.* Экспериментальные методы ядерной физики : - : НГТУ, 2010. - 219 с.

в) программное обеспечение и Интернет-ресурсы:

1. The Gaudi Framework, <http://proj-gaudi.web.cern.ch>
2. ROOT - A Data Analysis Framework. <http://root.cern.ch>
3. Geant4: a toolkit for the simulation of the passage of particles through matter. <http://geant4.cern.ch>
4. NOSQL Databases. <http://nosql-database.org>
5. СУБД PostgreSQL. <http://www.postgresql.org/docs>
6. СУБД MySQL. <http://dev.mysql.com/doc/>
7. СУБД Oracle. <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>
8. Документация Globus-Toolkit. <http://www.globus.org/toolkit>
9. NVIDIA CUDA Zone, <http://www.nvidia.com/cuda>

## 9. Материально-техническое обеспечение дисциплины

Лекционные занятия проводятся в аудитории, оборудованной мультимедийным проектором и компьютером, необходимых для презентации электронного варианта лекций и проведения компьютерных демонстраций.

Практические занятия проводятся в терминальном классе, оборудованном персональными компьютерами. Для занятий используется только свободно-распространяемое программное обеспечение (ОС LINUX, СУБД Postgres или MySQL, пакет ROOT и т.п.)