

Утверждено
Ректор Московского
государственного технического
университета им. Н.Э.Баумана

_____ Федоров И.Б.
«25» января 2010 г.

**Примерная
основная образовательная программа
высшего профессионального образования**

Направление подготовки

_____ 231000 Программная инженерия _____

утверждено приказом Минобрнауки России от 17 сентября 2009 г. № 337.

Квалификация (степень) выпускника _____ магистр _____

Нормативный срок освоения программы _____ 2 (два) года _____

Форма обучения - очная.

ФГОС ВПО утвержден приказом Минобрнауки России от 09.11.2009 г. № 543

Выпускник должен обладать следующими **общекультурными компетенциями (ОК):**

должен демонстрировать:

- способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень (ОК- 1);
- способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности (ОК- 2);
- умение свободно пользоваться русским и иностранным языками как средством делового общения (ОК -3);
- использование на практике умения и навыки в организации исследовательских и проектных работ, в управлении коллективом (ОК- 4);
- способность проявлять инициативу, в том числе в ситуациях риска брать на себя всю полноту ответственности (ОК- 5);
- способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности (ОК- 6);
- способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы) (ОК- 7);

Выпускник должен обладать следующими **профессиональными компетенциями (ПК):**

Научно-исследовательская деятельность

должен демонстрировать:

- умение отбирать и разрабатывать методы исследования объектов профессиональной деятельности на основе общих тенденций развития программной инженерии (ПК-1);
- умение проводить анализ, синтез, оптимизацию решений с целью обеспечения качества объектов профессиональной деятельности (ПК-2);
- умение организовывать самостоятельную и коллективную научно-исследовательскую работу (ПК-3).

Аналитическая деятельность

должен демонстрировать:

- умение планировать, управлять и контролировать выполнение требований (ПК-4);
- умение выполнять оценки степени трудности, рисков, бюджета и времени в течение выполнения проекта, осуществлять контроль рабочего графика (ПК-5).

Проектная деятельность

должен демонстрировать:

- способность к проектной деятельности в профессиональной сфере на основе системного подхода, умение строить и использовать модели для описания и прогнозирования различных явлений, осуществлять их качественный и количественный анализ (ПК-6);
- умение формировать технические задания и способность руководить разработкой программного обеспечения (ПК-7);
- умение оценить и выбрать методологию проектирования объектов профессиональной деятельности (ПК-8).

Технологическая деятельность

должен демонстрировать:

- умение применять современные технологии разработки программных комплексов с использованием автоматизированных систем планирования и управления, осуществлять контроль качества разрабатываемых программных продуктов (ПК-9).

Производственная деятельность

должен демонстрировать:

умение планировать и осуществлять руководство процессом разработки программного обеспечения (ПК-10).

Педагогическая деятельность

должен демонстрировать:

готовность использовать современные психолого-педагогические методы в профессиональной деятельности (ПК-11);

способность использовать педагогические приемы, принципы обучения и аттестации пользователей программного продукта при организации обучения (ПК-12);

навыки подготовки и проведения учебных занятий по дисциплинам направления «Программная инженерия» (ПК-13).

Организационно-управленческая деятельность

должен демонстрировать:

способность рассчитывать и оценивать условия и последствия принимаемых организационно-управленческих решений (ПК-14);

умение разработать техническое задание и провести технико-экономическое обоснование (ПК-15);

способность организовывать работу коллектива разработчиков программного продукта, умение осуществлять взаимодействие со смежниками (ПК-16).

Сервисно-эксплуатационная деятельность

должен демонстрировать:

умение осуществлять выбор технической и экономической моделей эволюции и сопровождения программного обеспечения (ПК-17).

ПРИМЕРНЫЙ УЧЕБНЫЙ ПЛАН

подготовки магистра по направлению подготовки «Программная инженерия»

Квалификация - магистр
Нормативный срок обучения – 2 года

Магистерская программа Разработка программно-информационных систем

№ п/п	Наименование дисциплин (в том числе практик)	Зачетные единицы	Академические часы	Примерное распределение по семестрам (количество семестров указывается в соответствии с нормативным сроком обучения, установленным ФГОС)						
				Трудоемкость по ФГОС	Трудоемкость	1-й семестр	2-й семестр	3-й семестр	4-й семестр	Форма промежуточной аттестации
						Количество недель (указывается количество недель по семестрам)				
1	2	3	4	5	6	7	8	9		
М.1 Общенаучный цикл		25	900	8	12	5				
	Базовая часть	8	288							
1.1	Философия и методология научного познания	4	144		X	X		3,3		
1.2	Методы научных исследований	4	144		X			э		
	Вариативная часть, в т.ч. дисциплины по выбору студента	17	612							
1.3	Цифровая обработка сигналов	6	216		X			э		
1.4	Теория параллельных вычислений	5	180	X				э		
	Дисциплины по выбору студента (2 из 3)	6	216	X		X				
	Алгоритмы распознавания									
	Математические основы верификации ПО									
	Иностранный язык									

М.2 Профессиональный цикл		35	1260	12	9	14	
	Базовая (общепрофессиональная) часть	10	360				
2.1	Конструирование компиляторов	4	144	X			э
2.2	Проектирование операционных систем	6	216			X	э
	Вариативная часть, в т.ч. дисциплины по выбору студента	25	900				
2.3	Процессы разработки программного обеспечения	5	180		X		з
2.4	Технологии высокопроизводительных вычислений	4	144		X		з
2.5	Разработка и реализация сетевых протоколов	4	144			X	з
2.6	Распределенные системы обработки информации	4	144	X			э
	Дисциплины по выбору студента (2 из 3)	8	288	X		X	
	Вычислительная лингвистика						
	Системы реального времени						
	Оценка качества программного обеспечения						
М.3 Практика и научно-исследовательская работа		48	1728				
М.4 Итоговая государственная аттестация		12	432				
Всего: <i>(указывается в соответствии с ФГОС)</i>		120	4320				

ПРИМЕРНАЯ ПРОГРАММА ДИСЦИПЛИНЫ «Цифровая обработка сигналов»

Рекомендуется для подготовки магистров
по направлению 231000 «Программная инженерия»

(Аннотация)

Цели освоения дисциплины

Дисциплина "Цифровая обработка сигналов" предназначена для студентов первого курса, обучающихся по направлению 231000 «Программная инженерия». В результате изучения курса студент должен знать непрерывное и дискретное преобразование Фурье (ДПФ) и его свойства, теорему о свертке, алгоритмы БПФ; особенности преобразования цветных изображений; понятие о вейвлет-преобразовании и его использовании для обработки сигналов; алгоритмы сжатия сигналов с потерями и без потерь, алгоритмы сжатия изображений JPEG, JPEG2000; алгоритмы классификации и распознавания образов.

Студент должен уметь: применять линейные и нелинейные фильтры для обработки сигналов; восстанавливать исходный сигнал из искаженного; выполнять классификацию известных объектов и распознавание (отнесение к классам) вновь предъявляемых объектов.

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Основные разделы дисциплины

Математическое представление сигналов. Принцип суперпозиции. Разложение сигнала по базисным функциям.

Дискретизация сигналов. Квантование сигналов. Теорема Котельникова и частота Найквиста. Равномерное и неравномерное квантование. Нелинейное предискажение сигнала. Практические аспекты дискретизации и квантования.

Непрерывные и дискретные преобразование Фурье (ДПФ) и Уолша (ДПУ). Теорема о свертке. Быстрые алгоритмы спектральных преобразований. Алгоритм быстрого преобразования Фурье (БПФ) с децимацией по времени. Сопряженный, сдвинутый и скользящий алгоритмы БПФ.

Линейные инвариантные к сдвигу фильтры (ЛИС-фильтры), Функция импульсного отклика и передаточная функция. Дискретное представление ЛИС-фильтров. Линейная и циклическая дискретная свертка. Алгоритм вычисления дискретной линейной свертки с использованием БПФ. Последовательная и секционная свертка. Классификация линейных фильтров. Фильтры с конечной (КИХ) и бесконечной (БИХ) импульсной характеристикой. Виды БИХ-фильтров. Фильтры Баттеруорта низких и высоких частот, полоснопропускающие и полоснозаграждающие фильтры. Виды КИХ-фильтров. Фильтры Ормсби и Поттера. Нелинейные алгоритмы обработки сигналов. Вариационный ряд, окрестность, ранг и срезка. Нелинейные алгоритмы сглаживания сигналов, выделение сигналов на фоне помех, обнаружения деталей и границ. Нелинейный алгоритм фильтрации импульсных помех. Стандартизация сигналов.

Восстановления исходного «идеального» изображения. Адаптивный фильтр Винера. Фильтрация аддитивной независимой помехи, фильтрация линейной помехи, вносимой передающей системой. Режекторный фильтр, препарирование сигналов. Линейные фильтры импульсных и полосовых помех. Метод регуляризации Тихонова. Некорректно поставленная задача. Параметр регуляризации, метод обобщенной невязки. Обзор методов «слепой»

ПРИМЕРНАЯ ПРОГРАММА ДИСЦИПЛИНЫ «Конструирование компиляторов»

Рекомендуется для подготовки магистров
по направлению 231000 «Программная инженерия»

(Аннотация)

Цели освоения дисциплины

Дисциплина "Конструирование компиляторов" предназначена для студентов первого курса, обучающихся по направлению 231000 «Программная инженерия». В результате изучения курса студент должен знать алгоритмы и методы построения лексических анализаторов; алгоритмы и методы построения возвратных и однопроходных синтаксических анализаторов; методы статического семантического анализа, основанные на атрибутивных грамматиках и атрибутивных трансляциях; алгоритмы и методы генерации промежуточного представления программы, основанные на атрибутивных грамматиках и атрибутивных трансляциях; алгоритмы и методы машинно-независимой оптимизации кода; методы генерации кода целевой машины; методы машинно-зависимой оптимизации кода.

Студент должен уметь разрабатывать лексические анализаторы; разрабатывать как однопроходные, так и многопроходные синтаксические анализаторы; применять методы обработки и нейтрализации синтаксических ошибок; применять методы статического семантического анализа для проверки типов и контекстно-зависимых условий; разрабатывать генераторы промежуточного представления программы, основанные на атрибутивных грамматиках и атрибутивных трансляциях; применять методы машинно-независимой оптимизации кода; разрабатывать генераторы кода целевой машины; применять методы машинно-зависимой оптимизации кода.

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Основные разделы дисциплины

Введение в компиляцию. Механизмы задания языков программирования. Фазы компилятора. Проходы компилятора. Контекст компилятора. Организация информации. Управление памятью.

Основные задачи, решаемые на этапе лексического анализа. Классификация методов построения лексических анализаторов.

Регулярные множества, регулярные выражения и праволинейные грамматики.

Алгоритм преобразования праволинейной грамматики в регулярные выражения. Язык расширенных регулярных выражений.

Конечные автоматы (КА). Связь между КА и языками, допускаемыми КА. КА-преобразователь как модель лексического анализатора. Минимизация КА.

Вопросы реализации лексических анализаторов. Организация информации о лексемах. Буферизация ввода. Обработка пробельных символов и комментариев. Представление диаграмм переходов. Способы управления программным моделированием КА-преобразователей. Реализация непрямого лексического анализа. Реализация прямого лексического анализа. Обработка ошибок в процессе лексического анализа.

ПРИМЕРНАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Проектирование операционных систем»

Рекомендуется для подготовки магистров
по направлению 231000 «Программная инженерия»

(Аннотация)

Цели освоения дисциплины

Дисциплина " Проектирование операционных систем " предназначена для студентов второго курса, обучающихся по направлению 231000 «Программная инженерия». В результате изучения курса студент должен знать основные понятия об архитектурном построении вычислительных и операционных систем (ОС), методах проектирования ОС различного назначения, видах ресурсов и методах их распределения в процессе функционирования ОС.

Студент должен уметь показать состав ОС различного назначения, принципы построения отдельных компонент; определить основные алгоритмы, используемые при построении компонент операционных систем; определить процедуры обслуживания ОС, и методы и средства управления ими.

Общая трудоемкость дисциплины составляет 6 зачетные единицы, 216 часов.

Основные разделы дисциплины

Эволюция ОС. Классификация ОС. Особенности алгоритмов управления ресурсами. Особенности аппаратных платформ. Особенности областей использования. Особенности методов построения.

Сетевые операционные системы. Структура сетевой операционной системы. Одноранговые сетевые ОС и ОС с выделенными серверами. ОС для рабочих групп и ОС для сетей масштаба предприятия.

Взаимодействие операционных систем в гетерогенных сетях. Понятия "internetworking" и "interoperability". Гетерогенность. Основные подходы к реализации взаимодействия сетей. Шлюзы. Мультиплексирование стеков протоколов. Использование магистрального протокола. Вопросы реализации. Сравнение вариантов организации взаимодействия сетей.

Управление процессами. Состояние процессов. Контекст и дескриптор процесса. Алгоритмы планирования процессов. Вытесняющие и невытесняющие алгоритмы планирования. Средства синхронизации и взаимодействия процессов. Нити.

Управление памятью. Типы адресов. Методы распределения памяти без использования дискового пространства. Распределение памяти фиксированными разделами. Распределение памяти разделами переменной величины. Перемещаемые разделы. Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти. Страничное распределение. Сегментное распределение. Странично-сегментное распределение. Свопинг. Иерархия запоминающих устройств. Принцип кэширования данных. Средства поддержки сегментации памяти. Сегментно-страничный механизм. Средства вызова подпрограмм и задач.

Управление вводом-выводом. Физическая организация устройств ввода-вывода. Организация программного обеспечения ввода-вывода. Обработка прерываний. Драйверы устройств. Независимый от устройств слой операционной системы. Пользовательский слой программного обеспечения.

ПРИМЕРНАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Теория и практика научных исследований»

**Рекомендуется для подготовки магистров
по направлению 231000 «Программная инженерия»**

(Аннотация)

Цели освоения дисциплины

Дисциплина "Теория и практика научных исследований" предназначена для студентов второго курса, обучающихся по направлению 231000 «Программная инженерия». Цель курса: дать представление о современной методологии и методах научных исследований, способах их организации и планирования, системе научных учреждений и подготовки кадров в ведущих странах мира, информационном обеспечении исследований, оформлении полученных результатов и подходах к оценке эффективности работы.

Задачи курса: дать развернутое изложение методов научной работы, ее планирования и организации в масштабах отдельного исследователя, коллектива и государства; выработать навыки подбора, анализа и обработки научной информации по теме исследования; научить формулировать цель и задачи исследования, планировать и проводить эксперимент (физический или вычислительный), обрабатывать результаты экспериментов, оценивать их достоверность; рассмотреть методы сопоставления теории (концепции, рабочей гипотезы) и эксперимента и основы формулирования научных выводов; изложить комплекс вопросов, относящихся к составлению отчетов, докладов и статей по результатам научного исследования.

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Основные разделы дисциплины

Общие сведения о науке и научных исследованиях. Основные определения и понятия. Определение науки и научного знания. Цель науки. Краткое введение в историю науки и принципы ее развития. Преемственность, дифференциация, специализация и интеграция наук. Классификация наук. Понятие о теориях, аксиомах, гипотезах, методах, методиках и методологии. Лженаука и мистика. Эмпирический и теоретический методы познания. Логический метод в науке. Темпы накопления научных знаний. Превращение науки в основную производительную силу общества. Движущие силы науки.

Классификация НИР. Место НИР в системе накопления знаний. Структурирование научных исследований по видам связи с производством, по источникам финансирования, по длительности разработки, по целевому назначению и т.д. Основные этапы научной работы.

Организация науки. Система научных учреждений страны и развитых государств мира. Система подготовки, аттестации и повышения квалификации научных и научно-педагогических кадров. Ученые степени и звания. Авторитет и признание в науке.

Выбор темы, определение цели и задач научных исследований. Формулирование и оценка темы. Понятие о научном направлении, проблеме, теме, вопросе. Требования к теме исследования, актуальность, научная значимость, новизна, экономическая или иная эффективность и т.д. Оценка важности, перспективности и осуществимости темы. Понятие о методе экспертных оценок.

ПРИМЕРНАЯ ПРОГРАММА ДИСЦИПЛИНЫ «Философия и методология научного познания»

**Рекомендуется для подготовки магистров
по направлению 231000 «Программная инженерия»**

(Аннотация)

Цели освоения дисциплины

Целью изучаемой дисциплины является ознакомление студентов со структурой научного знания, с методами научного исследования, с функциями научных теорий и законов; расширение их мировоззренческого кругозора; выработка представлений о критериях научности и о требованиях, которым должно отвечать научное исследование и его результаты, обретение студентами-магистрами знаний и суммы компетенций по оптимальной организации учебной, исследовательской и научной деятельности, по формированию текущих и долговременных стратегий развития естественно-научных и инженерно-технических исследований.

Задачами дисциплины является изучение предмета, назначения и основных функции философии и методологии научного познания, формулирование и решение проблем, возникающих в ходе научно-исследовательской деятельности и требующих углубленных профессиональных знаний, знаний в области философии и методологии научного познания; основных идей и результатов логики и методологии науки, обучение выбору необходимых методов исследования, модификации существующих и разработка новых методов, исходя из целей конкретного научного исследования; специфики научного исследования применительно к естественнонаучным и инженерно-техническим исследованиям; форм и методов участия в разработке совместно с другими членами коллектива общих научных проектов, требующих образования в соответствующем направлении;

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Основные разделы дисциплины

Философия и наука: формы взаимодействия. Основные способы освоения действительности человеком, место в них науки. Специфика философского и научного мышления. Исторические этапы развития взаимодействия философии и науки. Наука как система с рефлексией. Формы взаимодействия философии и науки. Типы рефлексии ученого: парадигмальная, мировоззренческая, методологическая, метатеоретическая, историко-научная, научно-организационная. Роль методологии в развитии знания. Методологическая культура.

Наука как социокультурный феномен. Познание и наука. Познание как активный, творческий процесс. Основные этапы исторического развития концепции познания. Особенности научного знания. Естественнонаучное и гуманитарное знание. Идеалы и нормы науки. Основания науки: идеалы и нормы научного исследования (идеалы и нормы: доказательности и обоснования знаний, объяснения и описания знаний, построения и организации знаний), научная картина мира; философские идеи и принципы. Наука как единство знания и познания. Ученый в системе научного производства. Наука в системе культуры.

Методы науки и их роль в познании. Понятие научного метода, его основные свойства. Классификация методов научного познания. Историчность научного метода. Границы

