

# Анализ бакалаврской образовательной программы по направлению «Информационные технологии»

Разработчик образовательной программы: Южный Федеральный Университет, факультет математики, механики и компьютерных наук.

Название образовательной программы: «Информационные технологии», тип: бакалаврская.

Название квалификации по окончании программы согласно диплому: бакалавр информационных технологий.

## Данные о программе

- место преподавания программы: Южный Федеральный Университет, факультет математики, механики и компьютерных наук;
- принадлежность: профессиональная;
- обычное время обучения: 4 года;
- количество кредитов ECTS: 246,4 кредита;
- модули (краткое описание):
  - модуль ГСЭ — гуманитарные и социально-экономические дисциплины;
  - модуль ЕН — математические и естественно-научные дисциплины;
  - модуль ОПД — общепрофессиональные дисциплины направления;
  - модуль СД — специальные дисциплины;
  - модуль ФД — факультативные дисциплины;

Курсы, относящиеся к различным модулям, за исключением ГСЭ разбиты более детально на следующие подмодули:

- *теоретическая и прикладная информатика*, куда входят дисциплины, связанные с дискретной математикой и алгеброй и их приложениями,

например, к проектированию формальных языков и распараллеливанию вычислений, а также классические разделы информатики, такие как базы данных;

- *системы и сети*, куда входят дисциплины, лежащие на стыке компьютерной инженерии и информатики, такие как операционные системы и сети, аппаратное обеспечение компьютера;
- *прикладная математика и программирование*, куда входят дисциплины, изучающие языки и технологии программирования, а также вычислительные аспекты и компьютерное моделирование;

- количество учебных мест: 25 бюджетных мест;
- наличие преподавательского состава: задействованы сотрудники факультета математики, механики и компьютерных наук Южного Федерального Университета с кафедр алгебры и дискретной математики, вычислительной математики и математической физики, информатики и вычислительного эксперимента, прикладной математики и программирования;
- плата за обучение: не взимается (для студентов, поступивших на бюджетные места на конкурсной основе);
- целевая аудитория: лица, имеющие среднее (полное) общее образование, а также выпускники техникумов, профиль которых связан с информационными технологиями;
- форма обучения: стационарная.

## **История**

Направление подготовки «Информационные технологии» существует на факультете математики, механики и компьютерных наук с 2005 года, в 2008/2009 учебном году состоялся первый выпуск бакалавров. Отчасти именно в связи с появлением этого направления факультет получил своё нынешнее имя (до этого он назывался механико-математическим). Руководителем бакалавриата ИТ является заведующий кафедрой Информатики и вычислительного эксперимента, профессор, д.ф.-м.н. В.С. Пилиди

Направление организовано в основном при поддержке трёх кафедр факультета: Алгебры и дискретной математики, Информатики и вычислительного эксперимента, Прикладной математики и программирования. Перечисленные кафедры имеют непосредственное отношение к развитию информатики в нашем регионе. Так, с именами людей, стоявших у истоков кафедр ПМП и ИВЭ, Л.А. Чикина и И.А. Николаева, связаны большие достижения в подготовке специалистов по программированию и информатике на юге страны. Нынешний заведующий кафедрой АДМ, Б.Я. Штейнберг, является руководителем одного из крупнейших в стране проектов по созданию оптимизирующих и распараллеливающих компиляторов.

## **Требования к бакалавру информационных технологий**

Бакалавр информационных технологий должен обладать следующими компетенциями:

- общекультурными:
  - способность выстраивать и реализовывать траектории интеллектуального, культурного, нравственного, физического и профессионального саморазвития и самосовершенствования;
  - уважительно и бережно относиться к историческому наследию и культурным традициям;

- толерантно воспринимать социальные и культурные различия;
- понимать и анализировать мировоззренческие, социально и личностно значимые философские проблемы;
- общепрофессиональными:
  - способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий;
  - способность профессионально решать задачи производственной и технологической деятельности с учетом современных достижений науки и техники, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования; разработку математических, информационных и имитационных моделей по тематике выполняемых исследований; создание информационных ресурсов глобальных сетей, образовательных контентов, прикладных баз данных; разработку тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; разработку эргономичных человеко-машинных интерфейсов;
  - способность разрабатывать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов систем информационных технологий, а также методы и механизмы оценки и анализа функционирования средств и систем информационных технологий; способность разработки проектной и программной документации, удовлетворяющей нормативным требованиям.

Профессиональные компетенции кроме вышеперечисленного включают в себя:

- научно-исследовательскую деятельность:
  - способность понимать и применять в исследовательской и прикладной деятельности современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий, способность использовать современные инструментальные и вычислительные средства;
  - способность в составе научно-исследовательского и производственного коллектива решать задачи профессиональной деятельности;
  - способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в сети Интернет, способность взаимодействовать и сотрудничать с профессиональными сетевыми сообществами и международными консорциумами, отслеживать динамику развития выбранных направлений области информационных технологий.
- производственно-технологическую деятельность:
  - способность применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные и вычислительные средства;
  - способность профессионально владеть базовыми математическими знаниями и информационными технологиями, эффективно применять их для решения научно-технических задач и прикладных задач,

связанных с развитием и использованием информационных технологий;

- способность осуществлять на практике современные методологии управления жизненным циклом и качеством систем, программных средств и сервисов информационных технологий;
- знание и следование в жизни кодекса профессиональной этики;
- способность формировать суждения о значении и последствиях своей профессиональной деятельности с учетом социальных, профессиональных и этических позиций;

● организационно-управленческую деятельность:

- способность реализовывать процессы управления качеством производственной деятельности, связанной с созданием и использованием систем информационных технологий, осуществлять мониторинг и оценку качества процессов производственной деятельности;
- способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы;
- способность осуществлять мониторинг за соответствием производственных процессов требованиям систем контроля окружающей среды и безопасности труда.

## **Информационно-методическое обеспечение учебного процесса**

Преподавателями направления было подготовлено большое количество методических указаний для поддержки учебных курсов направления «Информационные технологии». Они активно используются в учебном процессе.

Специально для информационно-методической поддержки курсов был открыт информационный портал по адресу <http://it.mmcs.sfedu.ru>. На нём размещаются материалы, связанные с учебными курсами: групповые и индивидуальные задания, раздаточные материалы для лекций, экзаменационные программы. Преподаватели участвуют в форуме портала и отвечают на вопросы студентов не только по обязательным учебным курсам программы, но и по факультативным и семинарским встречам. Наиболее активные студенты поддерживают электронные конспекты курсов по самым значимым ИТ-курсам с использованием вики-движка портала.

С недавнего времени для электронной поддержки курсов развёрнут система управления обучением Moodle по адресу <http://edu.mmcs.sfedu.ru>. В частности, на неё переведены курсы по основам программирования, компьютерным сетям, распределённым системам, основам разработки пользовательского интерфейса, платформе .NET. Moodle является результатом глубоких исследований в области педагогики (реализацией так называемой «педагогика социального конструкционизма») и уже после первых опытов использования позволила повысить качество и мобильность обеспечения методической поддержки образовательных курсов.

Ряд учебных курсов направления «Информационные технологии» ведётся с использованием мультимедийного оборудования — преподавателями подготовлены и используются презентации к лекциям по курсам веб-программирования, операционных систем, распределённых систем, моделированию информационных процессов. С ними можно познакомиться на вышеупомянутых ресурсах интернета.

# I. Цели

## I.a) описание целей

### 1. Цели образовательной программы

Образовательная программа по направлению «информационные технологии» (computing) имеет несколько целей, ориентированных на различные аспекты жизни общества. В самом общем виде их можно разделить на две группы: академические и профессиональные. В академической сфере программа направлена на подготовку ученых в области теоретической и прикладной информатики (computer science), способных к продвижению российской и мировой науки, на базе существующих и признанных во всём мире достижений и установившихся практик. Выпускник этой образовательной программы должен быть способным к продолжению обучения в магистратуре по направлениям:

- 511900 Информационные технологии,
- 510200 Прикладная математика и информатика,
- 552800 Информатика и вычислительная техника

и близким к ним направлениям, а также а также в аспирантуре преимущественно по научным специальностям:

- 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей;
- 05.13.12 Системы автоматизации проектирования (по отраслям);
- 05.13.13 Телекоммуникационные системы и компьютерные сети;
- 05.13.15 Вычислительные машины и системы;
- 05.13.17 Теоретические основы информатики;
- 05.13.18 Математическое моделирование, численные методы и комплексы;



- 05.13.19 Методы и системы защиты информации, информационная безопасность.

Что касается целей, связанных с подготовкой профессионалов, которые сразу же по завершении программы способны встроиться в существующую индустриальную среду, то здесь очевидна ориентация на подготовку специалистов в создании и поддержке информационных систем и сервисов, с использованием передовых техник и технологий прикладного программирования и администрирования вычислительных комплексов. Фундаментальные знания, которые закладывает программа, способствуют такому развитию студентов, при котором они способны системно подходить к появляющимся прикладным задачам и, что является особо актуальным в этой области, оперативно осваивать новые технологии, подстраиваться под быстро меняющиеся в индустрии тренды. Одна из главных целей в подготовке по этой специальности, поставленных в данной программе, связать глубокие теоретические достижения науки информатики и повсеместно принятые на сегодняшний день индустриальные практики.

В процессе прохождения программы студентам могут быть переданы базовые знания, относящиеся к фундаментальным разделам информатики, таким как теория алгоритмов и вычислимости, а также прикладным разделам, связанным с конкретными языками и системами программирования. Научная база по математике и информатике, заложенная в программе, поможет студентам заняться собственной исследовательской работой в будущем, если они выберут этот путь для своей дальнейшей деятельности. Разделы знаний, особенно важные в сегодняшних реалиях, такие как проектирование пользовательских интерфейсов и базы данных, гарантируют востребованность выпускников на рынке труда. Интересным аспектом прикладных дисциплин, которые поддерживаются преподавателями в актуальном состоянии, является необходимость для студентов самостоятельно искать и знакомиться с

информацией по технологиям, появившимся в последнее время, так как для них зачастую не существует (русскоязычной) печатной литературы. Это, безусловно, вырабатывает важнейший навык по поиску и систематизации нужной информации. Использование современных сред программирования, к примеру, позволит студентам относительно легко приспособиться к профессиональной деятельности в области прикладного программирования, если они предпочтут прикладную деятельность научным изысканиям по окончании учебной программы.

Главной особенностью данной образовательной программы по сравнению с уже существующими в регионе является попытка на качественно новом уровне сочетать фундаментальные и прикладные дисциплины в области информационных технологий. Специальности, существующие в технических вузах и имеющие схожие цели в отношении профессиональной подготовки молодежи страны, не обладают такой научной базой по математическим дисциплинам и курсам по теоретической информатике, как например, математическая логика и теория графов. С другой стороны, существующие фундаментальные специальности не могут предложить курсов, эквивалентных по актуальности излагаемого материала и возможности его применения непосредственно после завершения студентом учебной программы.

Другой важной особенностью является преподавательский состав специальности и, в частности, его возрастной уровень. Некоторые базовые курсы, такие как операционные системы, компьютерные сети, основы и языки программирования сознательно отданы молодым преподавателям, способным, основываясь на классической литературе по этим темам, внедрять новые технологии непосредственно используемые на практике в наши дни, а также, развивать новые формы взаимодействия и привлечения студентов к процессу обучения, в том числе дистанционные. С другой стороны ряд фундаментальных курсов, такие как введение в дискретную математику, теория формальных

языков и конечных автоматов, ведут профессора и доценты с большим опытом преподавания этих дисциплин, а также научными исследованиями по этим и близким направлениям.

Поскольку информатизация является глобальным процессом, который охватывает все новые стороны человеческой жизнедеятельности, профессионал в области информационных технологий будет востребован во многих сферах производства. Регулирование информационных потоков и создание соответствующей инфраструктуры, основанной на новейших достижениях, является безусловно востребованным в современном обществе. С другой стороны, наука информатики требует всё большего объема исследований, чтобы оперативно отвечать на вызовы времени. Данная образовательная программа призвана удовлетворить эти потребности.

## **2. Цели факультета, вуза**

Южный Федеральный Университет призван стать одним из крупнейших научных и образовательных центров страны. Достижение этой цели немыслимо без подготовки специалистов в области информационных технологий, ставших настолько актуальными сегодня. Руководство университета всячески поддерживало создание нового направления подготовки по данной программе, осознавая необходимость выпуска специалистов в этой области, а также заложения основы для фундаментальных исследований в области информатики на Юге страны.

Что касается факультета, на котором начата подготовка по данной программе, то с одной стороны, факультет математики, механики и компьютерных наук (бывший механико-математический факультет) стал благодатной почвой для введения этой учебной программы благодаря своим давним традициям и сильным научным школам по математике и механике; с другой стороны, открытие такого рода направления не могло не состояться из-за давно назревавших трудностей с ориентацией старых специальностей на

слишком широкий круг дисциплин, связанных с прикладной математикой и информатикой. Последние годы стало ясно, что для обеспечения высококачественной подготовки в области информатики, нельзя обойтись теми фундаментальными курсами, которые читались на имевшихся отделениях. Добавлять такие курсы в имевшиеся учебные планы не было технической возможности.

Существовавшие образовательные программы было тяжело изменить в какую-либо сторону, потому что они отвечали запросам основных научных школ и развивающихся на факультете направлений исследований. Результаты этого были двоякими: во-первых, выпускники не имели достаточно прочной базы даже по прикладному программированию (что, разумеется, не является недостатком конкретных программ, но сужает спектр видов подготовки, предоставляемых университетом в целом, и является принципиальным), во-вторых, профессора и доценты, интересы которых лежали в области информатики и дискретной математики, не могли получить достаточно хорошо подготовленных по этим направлениям учеников-аспирантов. Таким образом, внедрение данной образовательной программы мотивировано не только желанием иметь еще одно направление подготовки по актуальной специальности, но и объективными причинами, созревшими в недрах факультета за последние годы.

Особенно важным представляется стратегическое разделение направлений подготовки «информационные технологии» (по данной образовательной программе) и «прикладная математика и информатика». Даже названия здесь отражают возможность перекрытия целей программ. Однако в реальности существует определённое разделение, которое пытались поддерживать составители более новой образовательной программы по информационным технологиям (конечно, в некоторых разделах перекрытие, тем не менее, происходит). В отношении фундаментальных курсов первая программа

ориентирована на классические разделы информатики и дискретной математики, вторая — на непрерывную математику. Первая готовит исследователей в области информатики, для которых вычисления являются предметом изучения, а вторая — ученых, использующих компьютер в прикладных (особенно вычислительных) задачах и моделировании, для которых вычисления это лишь средство достижения результата (например, создания соответствующей естественнонаучной или экономической модели) или проверки гипотез.

### **3. Другие цели (государства)**

Взятый нашим государством курс на создание инновационной экономики целиком зависит от качества и масштабов информатизации производства и других сфер деятельности. В этом смысле подготовка специалистов по данной образовательной программе является стратегически важным для страны аспектом её существования.

#### **1.6) целевая аудитория**

Формальные требования для допуска к данной учебной программе определяются существующими юридическими нормами по приёму на обучение в государственное высшее учебное заведение и составляют необходимость получения поступающим среднего образования и успешную сдачу вступительных испытаний. Серьёзным недостатком данной системы на сегодняшний день является отсутствие вступительных испытаний по информатике, которые, безусловно, должны быть введены для допуска к данной программе. В противном случае, преподаватели некоторых базовых дисциплин, как например, основы программирования, сталкиваются с сильной дифференциацией учащихся по уровню подготовки, что создает препятствие для формирования достаточно эффективного курса, который мог бы служить прочным фундаментом основных дисциплин программы, таких как компьютерная графика, технологии баз данных и других.

Среди поступающих особо выделяются прошедшие подготовку в специализированных учебных заведениях, таких как Ростовский-на-Дону государственный колледж радиоэлектроники, информационных и промышленных технологий (РГКРИПТ), Ростовский авиационный колледж, и Воскресная компьютерная школа (ВКШ), действующая на факультете. Их процент, к сожалению, не высок.

Как обычно, после введения новой программы, обучение по ней не сразу становится престижным и популярным среди поступающих. Первые наборы были откровенно слабыми. Однако эта ситуация быстро меняется. В этом году поступило двое выпускников ВКШ, прошедших трехлетний курс подготовки по прикладному программированию, и имеющих, таким образом, чрезвычайно прочный бэкграунд; один из поступивших регулярно участвовал в городских и областных олимпиадах по информатике и также показывает высокие результаты по освоению учебной программы.

В этом году конкурс на обучение по данной программе был самым высоким на факультете. Тем не менее, он составлял немногим более 1 человека на место. Как представляется, у этого есть две причины. Первая — объективная — связана с общей неблагоприятной демографической ситуацией в стране, приходящейся на последние годы. Вторая касается падения популярности факультета среди абитуриентов города. Направление подготовки по данной учебной программе, в том числе, должно отразить запросы определенного сектора молодежи и помочь факультету справиться со второй проблемой.

По данной образовательной программе не завершил обучения ещё ни один набор учащихся. Последний набор составил 26 человек, из них 6 девушек и 19 юношей, все они имеют среднее (полное) общее образование, 5 бывших учащихся ВКШ, один бывший учащийся Ростовского Авиационного колледжа.

## I.в) обеспечение целей, описанных в а)

Содержание данной программы в основном отвечает современному научному уровню. При её составлении учитывались программы ведущих вузов в этой области: Массачусеттского технологического института, Университета Беркли в Калифорнии, Амстердамского свободного университета. Кроме того, особое внимание было уделено следованию Рекомендациям по преподаванию информатики в университетах Computing Curricula международных организаций ACM и IEEE.

Одним из важнейших аспектов реализации целей программы является наличие и компетенции кадрового состава. Здесь присутствуют как преимущества, так и недостатки. Следует признать, что на факультете имеется довольно мало заслуженных специалистов по информатике (доцентов и профессоров). Особенно это сказалось на таких курсах как операционные системы и сети. В областях, относящихся к дискретной математике, ситуация складывается лучше: к преподаванию привлечены профессора и доценты кафедр алгебры и дискретной математики и информатики и вычислительного эксперимента. Однако и тут существуют определённые проблемы. Так, к работе по реализации данной программы не был привлечен заведующий кафедрой алгебры, профессор, д. т. н., возглавляющий крупную исследовательскую работу, связанную с автоматическим распараллеливанием и оптимизацией в компиляторах — одной из актуальнейших отраслей современной фундаментальной информатики.

Привлечение молодых специалистов для ведения основных курсов имеет очевидные недостатки (отсутствие опыта) и менее очевидные преимущества. К последним относятся больший контроль за соответствием программы ожиданиям составителей, возможность ориентации на признанные авторитеты в ИТ-образовании, готовность преподавателя знакомиться с международными требованиями, в том числе, изучать опыт зарубежных коллег, преодолевая

языковые. В случаях, когда курс имеет явную техническую окраску большим плюсом является совмещение преподавания с работой в ИТ-отрасли, это чаще всего возможно только в случае молодых кадров, и в данном случае это происходит, например, с курсом программной инженерии.

Интересным инструментом повышения студенческой активности стал Сайт информационных технологий факультета математики, механики и компьютерных наук ЮФУ [it.mmcs.sfedu.ru](http://it.mmcs.sfedu.ru), который функционирует с первого года существования направления подготовки по данной учебной программе и служит для электронной поддержки учебных курсов. В частности, на нём можно найти информацию о преподавательском составе, методические материалы, презентации к лекциям, программы курсов. Важным его элементом является раздел форума, где наиболее активные студенты общаются с преподавателями, получая дополнительную информацию, как непосредственно о читаемых курсах, так и об области знаний в целом.



## II. Сведения об учебных планах отдельных дисциплин

Название курса	Алгоритмы и анализ сложности
Ответственный за курс	Столяр А. М., доцент, к.ф.-м.н.
Лекторы	Столяр А. М
Преподаватели практических занятий	Столяр А. М
Начало чтения курса	Зимний семестр 2007 г.
Семестр	5
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.8
Обязательные требования	Основы дискретной математики, Основы программирования, Алгебра и геометрия.
Рекомендуемые требования	Нет.
Обязателен для курсов	Нет.
Описание курса	Курс знакомит с рядом фундаментальных алгоритмов и современных методов их исследования, а также, оценки их теоретической сложности.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -формализации алгоритмических проблем; -использованию нотации, принятой при анализе и построении алгоритмов; -анализу как набора классических алгоритмов, так и встречающихся в практике решений частных алгоритмических задач; -построению алгоритмов с использованием стандартных методов; -доказательству корректности алгоритмов для типичных задач.
Программа курса	1. Основные понятия и обозначения. 2. Понятие сложности. Классы сложности. 3. Тактика Разделяй и властвуй. 4. Решение рекуррентных уравнений. Оценка рекуррентных соотношений. 5. Алгоритм Штрассена. 6. НВ- и НВП-разложения матрицы. 7. Дискретное преобразование Фурье. 8. Быстрое преобразование Фурье. 9. Теоретико-числовые задачи. 10. Некоторые вычислительные модели. Их связь. 11. Представления некоторых стандартных структур данных.
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 57 часов в семестр, 3 часа в неделю соответственно. Всего: 108 часов
Формы обучения	Лекции, занятия в классе, домашняя работа,

	устные доклады.
Формы промежуточного и итогового контроля	Устный доклад, письменные домашние задания, экзамен.
Требования к прохождению курса	Регулярное выполнение домашних работ, активное участие в практических занятиях, письменный экзамен
Литература	1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979 2. Ахо А., Хопкрофт Д., Ульман Дж. Структуры данных и алгоритмы. — М.: Издательский дом “Вильямс”, 2000. 3. Кузюрин Н. Н., Фомин С. А. Эффективные алгоритмы и сложность вычислений. Учебное пособие РАН. Институт системного программирования.
Примечания	

Название курса	Интеллектуальные системы
Ответственный за курс	Пучкин М.В., асс.
Лекторы	Пучкин М.В.
Преподаватели практических занятий	Пучкин М.В.
Начало чтения курса	Весенний семестр 2009 г.
Семестр	1
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3,8
Обязательные требования	Основы программирования, Языки программирования, Основы дискретной математики.
Рекомендуемые требования	Теория автоматов и формальных языков, Архитектура вычислительных систем, Структуры данных и алгоритмы, Технологии баз данных
Обязателен для курсов	Нет
Описание курса	В курсе излагаются история развития и современные концепции теории искусственного интеллекта, модели и методы представления знаний. Рассматриваются различные частные методы ИИ, такие, как нейронные сети, распознавание образов и автоматическое доказательство теорем.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -Пониманию базовых концепций интеллектуальных систем. -Пониманию основных методов решения задач с помощью ЭВМ, классификации и оценке сложности задачи и методов ее

	<p>решения.</p> <p>-Знанию основных моделей представления знаний в интеллектуальных системах, методах обработки знаний, их применимости в рамках различных моделей и на различных классах задач.</p> <p>-Разработке проекта и реализации экспертной системы для заданной предметной области.</p> <p>-Использованию систем проектирования и разработки искусственных нейронных сетей для решения широкого класса задач.</p>
Программа курса	<ol style="list-style-type: none"> <li>1. История искусственного интеллекта</li> <li>2. Социальные и этические аспекты ИИ</li> <li>3. Поиск решения задач</li> <li>4. Графы и пространства состояний</li> <li>5. Стратегии поиска: в ширину, в глубину, модификации</li> <li>6. Роль эвристик в решении задач</li> <li>7. Введение в теорию игр</li> <li>8. Задачи поиска с учетом ограничений</li> <li>9. Модели представления знаний</li> <li>10. Логическая модель представления знаний</li> <li>11. Доказательство теорем. Метод резолюции Байеса.</li> <li>12. Продукционная модель представления знаний</li> <li>13. Семантические сети</li> <li>14. Фреймовая модель представления знаний</li> <li>15. Нейронные сети</li> <li>16. Задачи классификации, распознавания образов</li> <li>17. Самообучающиеся системы</li> <li>18. Экспертные системы</li> </ol>
Распределение нагрузки студента	<p>Аудиторные часы: 45 часов в семестр (30 лекции, 15 практические), 3 часа в неделю соответственно.</p> <p>Самостоятельное изучение: 63 часа в семестр, 4,2 часа в неделю соответственно.</p> <p>Всего: 108 часов</p>
Формы обучения	Лекции, занятия в классе, домашняя работа
Формы промежуточного и итогового контроля	Домашняя работа по программированию, аудиторное тестирование, экзамен
Требования к прохождению курса	Регулярное выполнение домашних работ, активное участие в практических занятиях, письменный экзамен
Литература	<ol style="list-style-type: none"> <li>1. Джонс М. Т. Программирование искусственного интеллекта в приложениях – М.: ДМК Пресс, 2006.</li> </ol>

	<p>2. Люгер Д. Искусственный интеллект: стратегии и методы решения сложных проблем – М.: Вильямс, 2003.</p> <p>3. Рассел С., Норвиг П. Искусственный интеллект: современный подход – М.: Вильямс, 2006.</p> <p>4. Пучкин М.В. Интеллектуальные системы (ЭУП).</p>
Примечания	

Название курса	Компьютерная графика
Ответственный за курс	Русанова Я.М., доц., к.т.н.
Лекторы	Русанова Я.М.
Преподаватели практических занятий	Русанова Я.М., Пучкин М.В., асс.
Начало чтения курса	Осенний семестр 2007 г.
Семестр	5
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	2.5
Обязательные требования	Основы программирования, Языки программирования, Алгебра и геометрия.
Рекомендуемые требования	Структуры данных и алгоритмы
Обязателен для курсов	Доп. главы компьютерной графики
Описание курса	В курсе излагаются общие вопросы, методы и инструменты компьютерной графики.
Учебные цели	<p>В результате успешного усвоения данного курса студент должен быть способен к</p> <ul style="list-style-type: none"> <li>- Пониманию основных принципов и положений компьютерной графики</li> <li>- Созданию моделей трехмерных и двумерных объектов</li> <li>- Визуализации трехмерных объектов</li> <li>- Созданию графических приложений</li> <li>- Использованию в графических приложениях элементов событийного, объектно-ориентированного и многопоточного программирования</li> <li>- Владению различными современными технологиями, языками и библиотеками для работы с компьютерной графикой</li> </ul>
Программа курса	<ol style="list-style-type: none"> <li>1. Введение в компьютерную графику</li> <li>2. Растровые алгоритмы</li> <li>3. Геометрические преобразования</li> <li>4. Векторные алгоритмы на плоскости</li> <li>5. Векторные алгоритмы в пространстве</li> <li>6. Моделирование трехмерных объектов</li> <li>7. Концепции трехмерного наблюдения</li> <li>8. Представление трехмерных объектов</li> <li>9. Визуализация трехмерных объектов</li> <li>10. Построение реалистических изображе-</li> </ol>

	<p>ний</p> <p>11. Компьютерная анимация</p> <p>12. Форматы графических файлов</p>
Распределение нагрузки студента	<p>Аудиторные часы: 34 часа в семестр, 2 часов в неделю соответственно.</p> <p>Самостоятельное изучение: 34 часа в семестр, 4 часа в неделю соответственно.</p> <p>Всего: 68 часов</p>
Формы обучения	Лекции, занятия в компьютерном классе, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальные задания, зачет
Требования к прохождению курса	Выполнение индивидуальных заданий, активное участие в занятиях в компьютерном классе, зачет
Литература	<p>1. Роджерс Д., Адамс Дж. Математические основы компьютерной графики.</p> <p>2. Майкл Ласло. Вычислительная геометрия и компьютерная графика на C++.</p> <p>3. Шикин Е.В., Боресков А.В. Компьютерная графика. Динамика, реалистические изображения.</p> <p>4. Русанова Я.М. Алгоритмы компьютерной графики на плоскости и в пространстве</p>
Примечания	

Название курса	Доп. главы компьютерной графики
Ответственный за курс	Русанова Я.М., доц., к.т.н.
Лекторы	Русанова Я.М.
Преподаватели практических занятий	Русанова Я.М., Пучкин М.В., асс.
Начало чтения курса	Весенний семестр 2009 г.
Семестр	8
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	2.6
Обязательные требования	Основы программирования, Языки программирования, Алгебра и геометрия, Компьютерная графика
Рекомендуемые требования	Структуры данных и алгоритмы, Современное программирование
Обязателен для курсов	Нет.
Описание курса	В курсе излагаются общие вопросы, методы и инструменты компьютерной графики.
Учебные цели	<p>В результате успешного усвоения данного курса студент должен быть способен к</p> <ul style="list-style-type: none"> <li>- Использованию основных возможностей API: DirectX, Managed DirectX, OpenGL</li> <li>- Использованию высокоуровневого языка шейдеров</li> </ul>

	- Созданию полнофункциональных мультимедийных приложений
Программа курса	<ol style="list-style-type: none"> <li>1. Введение в DirectX</li> <li>2. Введение в OpenGL</li> <li>3. Технология рендеринга</li> <li>4. Mesh-объекты</li> <li>5. Использование ресурсов</li> <li>6. Введение в программируемый конвейер</li> <li>7. Основы программирования игр</li> <li>8. Язык шейдеров</li> <li>9. Добавление сетевых возможностей</li> </ol>
Распределение нагрузки студента	<p>Аудиторные часы: 34 часа в семестр, 2 часов в неделю соответственно.</p> <p>Самостоятельное изучение: 34 часа в семестр, 4 часа в неделю соответственно.</p> <p>Всего: 68 часов</p>
Формы обучения	Лекции, занятия в компьютерном классе, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальные задания, семинары, зачет
Требования к прохождению курса	Выполнение индивидуальных заданий, активное участие в семинарах, зачет
Литература	<ol style="list-style-type: none"> <li>1. Том Миллер DirectX 9 с управляемым кодом</li> <li>2. Джим Адамс DirectX. Продвинутая анимация</li> <li>3. Херн Бейкер Компьютерная графика и стандарт OpenGL</li> <li>4. Френсис Хилл OpenGL. Программирование компьютерной графики</li> </ol>
Примечания	

Название курса	Компьютерные сети
Ответственный за курс	Гуда С.А., асс.
Лекторы	Гуда С.А.
Преподаватели практических занятий	Гуда С.А.
Начало чтения курса	Зимний семестр 2009 г.
Семестр	4
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3,8
Обязательные требования	Основы программирования, Языки программирования
Рекомендуемые требования	Операционные системы
Обязателен для курсов	Распределенные системы Сетевое администрирование
Описание курса	В курсе излагаются основные сведения о компьютерных сетях, сети Интернет
Учебные цели	<p>В результате успешного усвоения данного курса студент должен</p> <p>-Понимать принципы сетевого</p>

	<p>взаимодействия компьютеров</p> <ul style="list-style-type: none"> <li>-Знать многоуровневые сетевые модели и стеки протоколов</li> <li>-Знать общую организацию сети Интернет</li> <li>-Знать основы веб-технологий</li> <li>-Уметь создавать сетевые приложения</li> <li>-Знать принципы маршрутизации в сети</li> <li>-Знать основы сетевой безопасности</li> </ul>
Программа курса	<p>1) Базовые понятия компьютерных сетей, структура сети Интернет.</p> <p>2) Многоуровневая сетевая архитектура</p> <p>3) Протоколы и службы прикладного уровня. DNS.</p> <p>4) WWW: HTTP, HTML, CGI. Клиенты и серверы.</p> <p>5) WWW: CSS, объектная модель документа DOM, JavaScript.</p> <p>6) Электронная почта, FTP, передача мультимедиа информации по сети</p> <p>7) VoIP, пиринговые сети. Введение в сетевое программирование.</p> <p>8) Интерфейс сокетов, протоколы TCP/UDP.</p> <p>9) Сетевой уровень: IP-адреса, DHCP, маршрутизация.</p> <p>10) Трансляция сетевых адресов, протоколы маршрутизации.</p> <p>11) Канальный уровень, кадрирование, управление ошибками, управление потоком,</p> <p>12) Протокол ARP. Основы технологии Ethernet: структура кадра, доступ к каналу.</p> <p>13) Физический уровень, оборудование сетей, среда передачи.</p> <p>14) Сетевая безопасность. Основы криптографии; алгоритмы шифрования; протоколы аутентификации.</p> <p>15) Электронная цифровая подпись; средства сетевой безопасности.</p>
Распределение нагрузки студента	<p>Аудиторные часы: 74 часов в семестр, 4 часа в неделю соответственно.</p> <p>Самостоятельное изучение: 22 часа в семестр, 1,3 часа в неделю соответственно.</p> <p>Всего: 96 часов</p>
Формы обучения	Лекции, занятия в компьютерном классе, семинары, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальные задания и семинары, экзамен
Требования к прохождению курса	Выполнение индивидуальных заданий, активное участие в семинарах, письменный экзамен

Литература	1. Э. Таненбаум. Компьютерные сети. 3 изд 2. Д. Куроуз, К. Росс. Компьютерные сети. Многоуровневая архитектура Интернета. 3. В. Г. Олифер, Н. А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. 3е изд. 2006г.
Примечания	

Название курса	Математическая логика и теория алгоритмов
Ответственный за курс	Ячменёва Н.Н., асс.
Лекторы	Ячменёва Н.Н.
Преподаватели практических занятий	Ячменёва Н.Н.
Начало чтения курса	Зимний семестр 2006 г.
Семестр	3
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	4.4
Обязательные требования	Основы дискретной математики
Рекомендуемые требования	Основы программирования
Обязателен для курсов	Неклассические логики
Описание курса	Данный курс является продолжением курса дискретной математики и рассматривает вопросы математических доказательств и оснований математики. Предметом логики являются логические исчисления, в том числе и предикатов. Также исследуются некоторые вопросы теории алгоритмов, изучающие общие свойства алгоритмов.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен: -формулировать задачи логического характера в рамках формальных систем ИС, ИВ и ИП. -выполнять преобразования логических формул с использованием схем тождественных преобразований. -проводить исследование логических формул для доказательства их свойств. -проводить доказательства в рамках аксиоматических систем. -формулировать и решать задачи, пользуясь машинами Тьюринга и рекурсивными функциями.
Программа курса	1. Предмет изучения и основные понятия. 2. Исчисления высказываний генценовского типа (ИС). Аксиомы и правила вывода. 3. Нормальные формы и эквивалентность в ИС. 4. Интерпретации ИС. 5. Непротиворечивость ИС. 6. Полнота, разрешимость и независимость



	<p>ИС.</p> <p>7. Исчисление высказываний гильбертовского типа. Теорема о дедукции.</p> <p>8. Эквивалентность ИС и ИВ.</p> <p>9. Алгоритмы проверки общезначимости и противоречивости в ИВ.</p> <p>10. Алгебраические системы. Термы, функции, предикаты.</p> <p>10. Исчисление предикатов (ИП).</p> <p>11. Нормальные формы и эквивалентность в ИП.</p> <p>12. Введение в теорию моделей.</p> <p>13. Теорема Гёделя о полноте ИП.</p> <p>14. Метод резолюций в ИП.</p> <p>15. Элементы теории алгоритмов: машины Тьюринга и рекурсивные функции. Тезис Чёрча.</p> <p>16. Неразрешимость ИП.</p> <p>17. Аксиомы формальной арифметики, теоремы о неразрешимости, о неполноте.</p> <p>18. Введение в теорию сложности алгоритмов.</p>
Распределение нагрузки студента	<p>Аудиторные часы: 68 часов в семестр, 4 часов в неделю соответственно.</p> <p>Самостоятельное изучение: 60 часов в семестр, 3.8 часов в неделю соответственно.</p> <p>Всего: 128 часов.</p>
Формы обучения	Лекции, занятия в классе, домашняя работа
Формы промежуточного и итогового контроля	Письменная домашняя работа, экзамен.
Требования к прохождению курса	Регулярное выполнение домашних работ, активное участие в практических занятиях, письменный экзамен.
Литература	<p>1. Ершов Ю.Л., Палютин Е.А. Математическая логика. — СПб, 2004.</p> <p>2. Судоплатов С.В., Овчинникова Е.В. Математическая логика и теория алгоритмов: Учебник. — М.: Инфра, 2004.</p> <p>3. Клини С.К. Математическая логика. — М.: КомКнига, 2007.</p>
Примечания	

Название курса	Операционные системы
Ответственный за курс	Брагилевский В.Н., асс.
Лекторы	Савельев В.А., доц., к.ф.-м.н., Брагилевский В.Н., асс.
Преподаватели практических занятий	Савельев В.А., Брагилевский В.Н.
Начало чтения курса	Осенний семестр 2007 г.
Семестр	5
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3,8

Обязательные требования	Основы программирования, Языки программирования
Рекомендуемые требования	Теория автоматов и формальных языков, Архитектура вычислительных систем, Структуры данных и алгоритмы
Обязателен для курсов	Распределенные системы
Описание курса	В курсе излагаются общие вопросы организации операционных систем, межпроцессное взаимодействие, подсистемы управления памятью и вводом выводом.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к - Пониманию принципов организации операционных систем. - Знанию механизмов синхронизации и межпроцессного взаимодействия. - Знанию принципов управления оперативной памятью и устройствами ввода/вывода. - Созданию многопоточных программ. - Созданию сценариев на языке командного интерпретатора.
Программа курса	1. Аппаратное обеспечение компьютеров. 2. Введение в операционные системы. 3. Подсистема управления процессами и потоками. 4. Механизмы синхронизации потоков. 5. Многопоточное программирование. 6. Межпроцессное взаимодействие. 7. Взаимоблокировки и голодание. 8. Планирование процессов и потоков. 9. Подсистема управления виртуальной памятью. 10. Подсистема управления вводом/выводом. 11. Файловые системы. 12. Безопасность операционных систем. 13. Операционные системы семейства Unix. 14. Операционные системы семейства Windows.
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 57 часов в семестр, 3.4 часа в неделю соответственно. Всего: 108 часов
Формы обучения	Лекции, занятия в компьютерном классе, семинары, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальные задания и семинары, экзамен
Требования к прохождению курса	Выполнение индивидуальных заданий, активное участие в семинарах, письменный

	экзамен
Литература	1. Э. Таненбаум. Современные операционные системы. 2. В. Столлингс. Операционные системы. 3. В.Н. Брагилевский, А.М. Пеленицын. Основы многопоточного и сетевого программирования. 4. В.Н. Брагилевский, А.В. Ермаков, А.М. Пеленицын. Операционные системы семейства Unix
Примечания	

Название курса	Основы дискретной математики
Ответственный за курс	Ерусалимский Я. М., проф., к.ф.-м.н.
Лекторы	Ерусалимский Я. М
Преподаватели практических занятий	Скороходов В.А., ст. преп., к.ф.-м.н.
Начало чтения курса	Зимний семестр 2005 г.
Семестр	1
Продолжительность	Два семестра
Зачетные единицы (Credit Points)	7.3
Обязательные требования	Нет
Рекомендуемые требования	Нет
Обязателен для курсов	Теория автоматов и формальных языков, Математическая логика и теория алгоритмов, Алгоритмы и анализ сложности, Теория конечных графов и её приложения, Интеллектуальные системы
Описание курса	Курс знакомит с основными понятиями дискретной математики, которые необходимы для занятия теоретической и прикладной информатикой.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -мыслению в категориях базовых дискретных структур и объектов; -решению простейших алгоритмических задач с использованием аппарата дискретной математики; -синтезу простейших алгоритмов над объектами дискретной математики; -определению области применимости методов дискретной математики;
Программа курса	1. Алгебра высказываний. 2. Теория Р-К схем и схем из функциональных элементов. 3. Алгебра предикатов. 4. Алгебра множеств. 5. Отображения. 6. Отношения. 7. Комбинаторика.

	8. Булевы функции. 9. Теория алгоритмов. 10. Элементы теории графов.
Распределение нагрузки студента	Аудиторные часы: 51 час первый и 54 второй семестр, 3 часа в неделю соответственно. Самостоятельное изучение: всего 99 часов, 3 часа в неделю соответственно. Всего: 204 часов
Формы обучения	Лекции, занятия в классе, домашняя работа
Формы промежуточного и итогового контроля	Письменная домашняя работа, зачет, экзамен
Требования к прохождению курса	Регулярное выполнение домашних работ, успешная сдача контрольных работ, письменный экзамен
Литература	1. Нефедов В.Н., Осипова В.А. Курс дискретной математики: М.: Изд-во МАИ. 1992. 264 с. 2. Яблонский С.В. Введение в дискретную математику: М.: Наука. 1979 3. Ерусалимский Я.М. Дискретная математика: теория, задачи, приложения: М.: Вузовская книга, 1998, 280 с.
Примечания	

Название курса	Основы программирования
Ответственный за курс	Михалкович С.С., доц., к.ф.-м.н.
Лекторы	Михалкович С.С.
Преподаватели практических занятий	Брагилевский В.Н., асс.
Начало чтения курса	Зимний семестр 2005 г.
Семестр	1
Продолжительность	Два семестра
Зачетные единицы (Credit Points)	7.3
Обязательные требования	Нет
Рекомендуемые требования	Нет
Обязателен для курсов	Структуры данных и алгоритмы, практический курс программирования
Описание курса	
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -Пониманию как базовых концепций императивных и объектно-ориентированных языков программирования, так и важных программных средств на этих языках. -Пониманию базовых концепций программирования в логических и функциональных языках. -Знанию основных типов данных и их реализации с помощью различных программных средств. -Созданию небольших программ и

	документированию их в соответствии с общими программистскими соглашениями. -Знанию основных путей описания программных языков. -Тестированию базовых программ.
Программа курса	<ol style="list-style-type: none"> <li>1. Алгоритмы и алгоритмические языки программирования.</li> <li>2. Введение в язык программирования Паскаль.</li> <li>3. Понятие типа данных. Совместимость типов по присваиванию.</li> <li>4. Понятие подпрограммы.</li> <li>5. Модульность и отдельная компиляция.</li> <li>6. Обработка массивов и строк.</li> <li>7. Файлы.</li> <li>8. Введение в язык программирования C++.</li> <li>9. Встраиваемые функции.</li> <li>10. Перегрузка имён функций.</li> <li>11. Особенности отдельной компиляции в C++.</li> <li>12. Пространства имён.</li> <li>13. Введение в ООП на Object Pascal.</li> <li>14. Указатели в ОР и в C++.</li> <li>15. Динамические структуры данных.</li> <li>16. Рекурсия.</li> </ol>
Распределение нагрузки студента	<p>Аудиторные часы: 34 часов в первый семестр, 2 часа в неделю соответственно. 36 часов во второй семестр, 2 часа в неделю соответственно.</p> <p>Самостоятельное изучение: 99 часов всего, 3 часа в неделю соответственно.</p> <p>Всего: 204 часов</p>
Формы обучения	Лекции, занятия в компьютерном классе, домашняя работа.
Формы промежуточного и итогового контроля	Лабораторные работы, экзамены
Требования к прохождению курса	Регулярное выполнение лабораторных работ, письменные экзамены.
Литература	<ol style="list-style-type: none"> <li>1. Фаронов В.В. Turbo Pascal 7.0.</li> <li>2. Страуструп Б. Язык программирования C++, 3-е изд.</li> <li>3. Вирт Н. Алгоритмы и структуры данных.</li> </ol>
Примечания	

Название курса	Программная инженерия
Ответственный за курс	Адигеев М.Г., ст. преп., к.т.н.
Лекторы	Адигеев М.Г.

Преподаватели практических занятий	Адигеев М.Г.
Начало чтения курса	Зимний семестр 2008 г.
Семестр	7
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.4
Обязательные требования	Основы программирования.
Рекомендуемые требования	Инновационный менеджмент.
Обязателен для курсов	Нет.
Описание курса	В курсе последовательно рассматриваются современные методики, применяемые при индустриальной разработке программного обеспечения.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -описанию жизненного цикла программного продукта (ПП) -использованию или быстрому освоению технологий и инструментов, применяемых на каждом этапе разработки ПП; -использованию базовых понятий и принципов проектного менеджмента на практике; -быстрой адаптации в новых коллективах разработчиков ПП, использующих современные технологии и процессы разработки ПП.
Программа курса	1. Введение в дисциплину. 2. Обзор методологий разработки ПО. 3. Управление командой разработчиков. 4. Работа с требованиями к ПО. 5. Планирование и проектирование архитектуры ПО. 6. Планирование и контроль процесса разработки ПО. 7. Методы управления рисками и сроками в процессе разработки ПО. 8. Организация и методы тестирования. 9. Разработка документации. 10. Развертывание и сопровождение программного продукта. 11. Обзор популярных методологий разработки: RUP, MSF, XP. 12. Обзор отраслевых стандартов.
Распределение нагрузки студента	Аудиторные часы: 48 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 48 часов в семестр, 3 часа в неделю соответственно. Всего: 96 часов
Формы обучения	Лекции, занятия в классе, домашняя работа.
Формы промежуточного и итогового контроля	Индивидуальное задание, зачёт.

Требования к прохождению курса	Выполнение индивидуального задания, устный зачёт.
Литература	1. Адигеев М.Г. Жизненный цикл программного обеспечения. Метод. указания. // Готовится к публикации. 2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. — СПб.: Символ-Плюс, 1999. 3. Бек К. Экстремальное программирование. — СПб.: Питер, 2002.
Примечания	

Название курса	Распределенные системы
Ответственный за курс	Брагилевский В.Н.
Лекторы	Брагилевский В.Н.
Преподаватели практических занятий	Брагилевский В.Н.
Начало чтения курса	Весенний семестр 2008 г.
Семестр	6
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3,6
Обязательные требования	Основы программирования, Языки программирования, Операционные системы, Компьютерные сети
Рекомендуемые требования	Структуры данных и алгоритмы
Обязателен для курсов	
Описание курса	В курсе излагаются общие вопросы организации распределенных систем, распределенные алгоритмы и архитектуры сетевых приложений.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к - Пониманию принципов организации распределенных систем. - Знанию основных распределенных алгоритмов. - Знанию основных видов распределенных систем. - Знанию основных архитектур сетевых приложений. - Созданию сетевых приложений.
Программа курса	1. Многопроцессорные системы 2. Архитектура сетевых приложений 3. Введение в распределенные системы 4. Проблемы распределенной синхронизации. 5. Системы именования. 6. Реализация распределенных транзакций. 7. Непротиворечивость данных и репликация. 8. Отказоустойчивость.

	9. Распределенные системы объектов 10. Веб-сервисы 11. Сетевые файловые системы 12. Распределенные системы согласования
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 51 час в семестр, 3 часа в неделю соответственно. Всего: 102 часа.
Формы обучения	Лекции, занятия в компьютерном классе, домашняя работа
Формы промежуточного и итогового контроля	Лабораторные работы и индивидуальные задания, экзамен
Требования к прохождению курса	Выполнение индивидуальных заданий, письменный экзамен
Литература	1. Э. Таненбаум. Распределенные системы. 2. Х.М. Дейтел, П.Дж. Дейтел, Д.Р. Чофнес. Операционные системы. 3. В.Н. Брагилевский, А.М. Пеленицын. Основы многопоточного и сетевого программирования.
Примечания	

Название курса	Современное программирование
Ответственный за курс	Абрамян М. Э., доц., к.ф.-м.н.
Лекторы	Абрамян М. Э.
Преподаватели практических занятий	Абрамян М. Э.
Начало чтения курса	Весенний семестр 2009 г.
Семестр	8
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.6
Обязательные требования	Основы программирования, Языки программирования, Технологии баз данных.
Рекомендуемые требования	XML-технологии.
Обязателен для курсов	Нет.
Описание курса	Курс излагает современные концепции и приемы программирования, реализованные для платформы .NET Framework.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -пониманию базовых концепций платформы .NET Framework и возможностей её применения в промышленном программировании. -использованию технологий платформы .NET Framework для решения стандартных задач прикладного программирования, таких как создание клиентов баз данных, сериализация данных и др.;



	-пониманию и использованию в подходящих случаях концепции динамической кодогенерации; -созданию и упаковке готовых программных продуктов.
Программа курса	1. Сериализация и десериализация объектов 2. Технология Interop и использование доменов приложений. 3. Механизм отражения (reflection) и его использование для динамической генерации кода. 4. Доступ к данным с помощью механизмов ADO.NET. 5. Средства для установки и конфигурирования приложений.
Распределение нагрузки студента	Аудиторные часы: 45 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 57 часов в семестр, 3 часа в неделю соответственно. Всего: 102 часа.
Формы обучения	Лекции, занятия в классе, домашняя работа
Формы промежуточного и итогового контроля	Домашняя работа по программированию, экзамен
Требования к прохождению курса	Регулярное выполнение домашних работ, письменный экзамен
Литература	1. Абрамян М. Э. Практикум по разработке Windows-приложений на платформе .NET 2.0. // Готовится к печати. 2. Microsoft Corporation. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. М.: Русская редакция, 2003. 3. Нортроп Т., Уилдермьюс Ш., Райан Б. Основы разработки приложений на платформе Microsoft .NET Framework. М.: Русская редакция, 2007.
Примечания	

Название курса	Современные web-технологии
Ответственный за курс	Русанова Я.М.
Лекторы	Русанова Я.М.
Преподаватели практических занятий	Русанова Я.М., Пучкин М.В.
Начало чтения курса	Весенний семестр 2008 г.
Семестр	6
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	2.5
Обязательные требования	Основы программирования, Языки программирования, Компьютерные сети
Рекомендуемые требования	Структуры данных и алгоритмы
Обязателен для курсов	Нет.

Описание курса	В курсе излагаются общие концепции и приемы web-программирования, современные клиентские и серверные web-технологии.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к <ul style="list-style-type: none"> <li>- Знанию и использованию существующих клиентских web-технологий</li> <li>- Знанию и использованию современных серверных web-технологий</li> <li>- Использованию современных языков для создания web-приложений</li> <li>- Созданию различных видов web-приложений, в частности, web-сервисов, сайтов, порталов</li> </ul>
Программа курса	<ol style="list-style-type: none"> <li>1. История и основные тенденции развития Web – технологий</li> <li>2. Принципы гипертекстовой разметки и каскадные таблицы стилей CSS</li> <li>3. Язык JavaScript и динамический HTML</li> <li>4. Средства создания и поддержки Web – приложений</li> <li>5. CGI - технологии</li> <li>6. Обзор языков программирования Perl и PHP</li> <li>7. Организация серверной части web-приложения и интеграция с базами данных</li> <li>8. Безопасность web-приложений</li> <li>9. Web-сервисы</li> </ol>
Распределение нагрузки студента	<p>Аудиторные часы: 34 часа в семестр, 2 часов в неделю соответственно.</p> <p>Самостоятельное изучение: 34 часа в семестр, 2 часа в неделю соответственно.</p> <p>Всего: 68 часов</p>
Формы обучения	Лекции, занятия в компьютерном классе, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальные задания, контрольные работы, зачет
Требования к прохождению курса	Выполнение индивидуальных заданий, активное участие в занятиях в компьютерном классе, успешное написание контрольных работ, зачет
Литература	<ol style="list-style-type: none"> <li>1. Ален Вайк, Джейсон Джиллиам. JavaScript. Полное руководство.</li> <li>2. Д. Котеров, А.Костарев. PHP5.</li> <li>3. Русанова Я.М. Клиентские web-технологии</li> </ol>
Примечания	

Название курса	Теория конечных графов и её приложения.
Ответственный за курс	Ерусалимский Я. М., проф., к.ф.-м.н.
Лекторы	Ерусалимский Я. М.
Преподаватели практических занятий	Ерусалимский Я. М.
Начало чтения курса	Весенний семестр 2008 г.
Семестр	6
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.8
Обязательные требования	Основы дискретной математики
Рекомендуемые требования	Алгоритмы и анализ сложности
Обязателен для курсов	Нет.
Описание курса	В курсе дается описание конечных графов, их основные характеристики, базовые алгоритмы на графах, их приложения.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -пониманию и свободному использованию фундаментальных понятий теории графов; - пониманию современной проблематики теории графов -изложению задач теории графов и методов их решения -применению основных теоретико-графовых алгоритмов -применению графовых моделей к различным областям науки.
Программа курса	1. Определение, примеры, способы задания графов. 2. Изоморфизм графов. 3. Локальные характеристики графов. Теорема Эйлера о рукопожатиях. 4. Пути, цепи, контуры и циклы. 5. Мосты и точки сочленения. 6. Деревья и леса. Основная теорема о деревьях и следствия из неё. 7. Задача о минимальном соединении. Алгоритм Краскала. 8. Раскраски графа. 9. Кратчайшие пути на графе. Алгоритм Дейкстры и его матричная реализация. 10. Лин. пространства над графами. 11. Пространства разрезов графа. 12. Устойчивые множества на графах. 13. Ядро графа. 14. Функции на графах. 15. Операции над графами. 16. Помеченные деревья.
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно.

	Самостоятельное изучение: 57 часа в семестр, 3 часа в неделю соответственно. Всего: 108 часов.
Формы обучения	Лекции
Формы промежуточного и итогового контроля	экзамен
Требования к прохождению курса	письменный экзамен
Литература	1. Харари. Теория графов. 1973. 2. Кристофидес Н. Теория графов. Алгоритмический подход. 1978. 3. Оре О. Теория графов. 1980.
Примечания	

Название курса	Технологии баз данных
Ответственный за курс	Чердынцева М. И., доц., к.т.н.
Лекторы	Чердынцева М. И.
Преподаватели практических занятий	Чердынцева М. И.
Начало чтения курса	Весенний семестр 2008 г.
Семестр	6
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.4
Обязательные требования	Алгебра и геометрия, Основы программирования
Рекомендуемые требования	Языки программирования
Обязателен для курсов	Структуры данных и алгоритмы, практический курс программирования
Описание курса	В курсе излагаются основные понятия и методы организации реляционных баз данных и манипулирования ими, а также описываются базовые подходы к проектированию реляционных баз данных.
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -определению места и необходимости применения базы данных в современных приложениях; -пониманию реляционной модели данных и ее реализации в распространенных СУБД; -построению ER–диаграмм; -проектированию баз данных; -нормализации баз данных; -обеспечению целостности данных в базе данных; -решению проблем отсутствующих значений в базе данных; -построению запросов на языке SQL.
Программа курса	1. Причины внедрения баз данных. 2. Реляционный подход, фундаментальные свойства отношений. 3. Реляционная база данных.

	4. Нормализация. 5. Модель «сущность–связь» (ER). 6. Индексы. 7. Первичные ключи. 8. Представления. 9. Системы безопасности и роли. 10. Транзакции. 11. Хранимые процедуры. 12. Триггеры. 13. Оператор SELECT. 14. Соединения, группировки, подзапросы. 15. Операторы модификации данных и объектов схемы базы данных.
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 45 часов в семестр, 3 часа в неделю соответственно. Всего: 96 часов
Формы обучения	Лекции, занятия в классе, домашняя работа
Формы промежуточного и итогового контроля	Индивидуальное задание, экзамен
Требования к прохождению курса	Выполнение индивидуального задания, письменный экзамен
Литература	1. Борри Х. Firebird: руководство разработчика баз данных. — СПб.: БХВ–Петербург, 2006. 2. Гарсия–Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. — М.:Издательский дом «Вильямс», 2004. 3. Грофф Дж., Вайнберг П. SQL: полное руководство. — К.: Издательская группа BHV, 1998.
Примечания	

Название курса	Теория автоматов и формальных языков
Ответственный за курс	Деундяк В.М., доц., к.ф-м.н.
Лекторы	Деундяк В.М.
Преподаватели практических занятий	Деундяк В.М.
Начало чтения курса	Весенний семестр 2008 г.
Семестр	4
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.8
Обязательные требования	Основы дискретной математики, Математическая логика и теория алгоритмов, Основы программирования, Алгебра и геометрия.
Рекомендуемые требования	Элементы общей алгебры, Языки программирования.
Обязателен для курсов	Моделирование информационных процессов, Методы трансляции.
Описание курса	Курс посвящён классическому разделу

	математической лингвистики и теоретической информатики, в котором рассматривается построение класса формальных вычислительных моделей, а также математическая теория формальных языков.
Учебные цели	<p>В результате успешного усвоения данного курса студент должен быть способен к</p> <ul style="list-style-type: none"> <li>-пониманию задач теории формальных моделей вычисления и синтаксического анализа;</li> <li>-применению математического аппарата для анализа формальных моделей вычислительных процессов и структур;</li> <li>-пониманию конструкций теории формальных языков, а также основных принципов и методов синтаксического анализа формальных языков (в т.ч. языков программирования);</li> <li>-классификации конкретных формальных языков;</li> <li>-описанию формальных языков, моделей вычислений, используемых для представления формальных языков.</li> </ul>
Программа курса	<ol style="list-style-type: none"> <li>1. Структуры общей алгебры.</li> <li>2. Дискретные системы и логические схемы, формальные языки и грамматики.</li> <li>3. Способы определения языков.</li> <li>4. Регулярные множества и конечные автоматы.</li> <li>5. Контекстно-свободные языки.</li> <li>6. Нормальные формы контекстно-свободных грамматик.</li> <li>7. Автоматы с магазинной памятью.</li> <li>8. Основные понятия перевода.</li> <li>9. Конечный преобразователь.</li> <li>10. Магазинный преобразователь.</li> <li>11. Эквивалентность конкретных классов языков конкретным типам преобразователей.</li> </ol>
Распределение нагрузки студента	<p>Аудиторные часы: 54 часов в семестр, 3 часа в неделю соответственно.</p> <p>Самостоятельное изучение: 54 часов в семестр, 3.2 часа в неделю соответственно.</p> <p>Всего: 104 часа</p>
Формы обучения	Лекции, занятия в классе, домашняя работа.
Формы промежуточного и итогового контроля	Домашние задания, в том числе, индивидуальные, экзамен.
Требования к прохождению курса	Регулярное выполнение домашних работ, активное участие в практических занятиях, письменный экзамен

Литература	1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции Т 1. Синтаксический анализ. — М.: Мир, 1978. 2. Хопкрофт Д., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений – М.: Вильямс, 2002. 3. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки, Программирование. — Киев: Наукова Думка, 1989. 4. Деундяк В.М. Элементы теории формальных языков — Ростов н/Д: ДГТУ, 1997.
Примечания	

Название курса	Языки программирования
Ответственный за курс	Михалкович С.С., доц., к.ф.-м.н.
Лекторы	Михалкович С.С.
Преподаватели практических занятий	Брагилевский В.Н., асс.
Начало чтения курса	Зимний семестр 2006 г.
Семестр	3
Продолжительность	Один семестр
Зачетные единицы (Credit Points)	3.8
Обязательные требования	Основы программирования
Рекомендуемые требования	Нет
Обязателен для курсов	Современное программирование, Современные web-технологии, Интеллектуальные системы, Операционные системы, Компьютерные сети
Описание курса	
Учебные цели	В результате успешного усвоения данного курса студент должен быть способен к -применению базовых концепций ООП; -эффективному использованию основных объектных моделей; -моделированию задач предметной области с помощью ОО-языков; -использованию механизма исключительных ситуаций в ОО-языках; -применению базовых средств стандартных библиотек языков C++/Java/Object Pascal.
Программа курса	1. Введение в Java. 2. Абстрактные типы данных (АТД). Классы. 3. Свойства в Object Pascal. 4. Объектная модель Java/OP/C++. 5. Реализация ряда контейнерных классов. 6. Моделирование. 7. Клонирование и присваивание. 8. Наследование.

	9. Присваивания в иерархиях. 10. Полиморфизм. 11. Исключения. 12. Перегрузка операций. 13. Знакомство с шаблонами C++. 14. Информация о типе во время выполнения в C++. 15. Потоки C++.
Распределение нагрузки студента	Аудиторные часы: 51 часов в семестр, 3 часа в неделю соответственно. Самостоятельное изучение: 57 часов в семестр, 3 часа в неделю соответственно. Всего: 108 часов
Формы обучения	Лекции, занятия в компьютерном классе, групповые задания, домашняя работа.
Формы промежуточного и итогового контроля	Лабораторные и групповые задания, экзамен.
Требования к прохождению курса	Регулярное выполнение лабораторных работ, письменный экзамен.
Литература	1. Мейер Б. Объектно-ориентированное конструирование программных систем. 2. Хорстман, Корнелл. Java 2, 7-изд. 3. Страуструп. Язык программирования C++, 3-е изд.
Примечания	