

## ОБЪЕМ АУДИТОРНОЙ УЧЕБНОЙ НАГРУЗКИ ПО БАЗОВОМУ КУРСУ «ИНФОРМАТИКА»

РАЗДЕЛ КУРСА	семестр	Часов в неделю		Всего часов аудиторных занятий по разделу
		Лекции	Практикум	
<b>1. Офисные технологии</b>	<b>1</b>		<b>16</b>	<b>16</b>
<b>2. Программирование</b>	<b>1-3</b>			<b>192</b>
<b>2.1. Основы программирования</b>	<b>1-2</b>	<b>60</b>	<b>60</b>	<b>120</b>
<b>2.2. Объектно-ориентированное программирование</b>	<b>3</b>	<b>44</b>	<b>28</b>	<b>72</b>
<b>3. Форматы данных и языки разметки</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>8</b>
<b>4. Компьютерные сети</b>	<b>4</b>	<b>10</b>	<b>10</b>	<b>20</b>
<b>5. Управление информацией</b>	<b>4</b>	<b>12</b>	<b>12</b>	<b>24</b>
<b>6. Компьютерная графика</b>	<b>4</b>	<b>6</b>	<b>6</b>	<b>12</b>

	Часов в неделю		Часов в семестр
	Лекции	Практика	
<b>1 семестр</b>	<b>2</b>	<b>3</b>	<b>80</b>
<b>2 семестр</b>	<b>2</b>	<b>2</b>	<b>56</b>
<b>3 семестр</b>	<b>3</b>	<b>2</b>	<b>80</b>
<b>4 семестр</b>	<b>2</b>	<b>2</b>	<b>56</b>
<b>Всего часов аудиторных занятий</b>			<b>272</b>

**ПРОГРАММА ДИСЦИПЛИНЫ**  
**«ИНФОРМАТИКА»**  
**ДЛЯ НЕПРОФИЛЬНЫХ СПЕЦИАЛЬНОСТЕЙ**  
**КЛАССИЧЕСКИХ УНИВЕРСИТЕТОВ**

**1. ОФИСНЫЕ ТЕХНОЛОГИИ**

*Цели: Приобретение навыков работы на персональном компьютере. Иметь представление о современных офисных пакетах программ и их характерных особенностях. Умение использовать офисные технологии в делопроизводстве. Умение оформлять научные отчеты и публикации. Навыки работы с различными данными: текстовой информацией, графическими изображениями, численными данными. Умение обработать и представить результаты экспериментов, опросов и другие подобные данные. Приобретение навыков работы в сети Интернет. Умение поиска нужной информации в глобальной сети.*

- 1.1. Принципы работы с компьютером. Операционная система. Работа с файловой системой. Окружение и стандартные программы.
- 1.2. Компьютерная графика, обработка графических изображений. Программы для работы с графическими изображениями.
- 1.3. Текстовый процессор. Оформление документа. Инструменты редактирования. Средства управления текстом и графикой. Создание документов с использованием формул и графических изображений.
- 1.4. Использование электронных таблиц для обработки данных. Рабочие листы. Инструменты редактирования. Расчеты. Построение графиков и диаграмм.
- 1.5. Презентационный процессор. Создание слайдов и презентаций различного назначения. Обмен данными и документами в офисных приложениях.
- 1.6. Пакеты прикладных программ для научных расчетов (численные методы, символьные вычисления).
- 1.7. Понятие глобальной сети Internet. Гипертекст. Браузер. Поисковые машины.
- 1.8. Электронная почта, основные функции почтового сервера и почтового клиента. Понятие телеконференции.

## 2. ПРОГРАММИРОВАНИЕ

*Цели:* Овладение основами алгоритмического мышления и умением сводить решение задачи к написанию алгоритма. Умение записывать алгоритмы на неформальном языке и владение языком практического программирования Си. Умение сознательно применять конструкции алгоритмического языка и доказывать правильность простейших программ по их тексту. Знание принципов устройства компьютера, понимание простейших программ в кодах и на языке Ассемблера. Знание основных вычислительных алгоритмов (численное интегрирование, нахождение корней функций, интерполяция). Знание основных структур данных и их реализаций, умение применять их в практических задачах. Знание элементарной теории, применяемой при работе со структурами данных: задачи поиска и сортировки, теоретические оценки снизу числа операций и оптимальные алгоритмы сортировки, хеширование, применение бинарных деревьев.

### 2.1. ОСНОВЫ ПРОГРАММИРОВАНИЯ

#### 2.1.1. Общее понятие алгоритма

Понятие алгоритма. Алгоритмические языки. Запись алгоритма на неформальном языке (псевдокоде). Элементы алгоритмического языка. Подпрограмма и функция, оператор "если", цикл "пока" и арифметический цикл.

Понятие переменной. Наиболее распространенные типы переменных: целые, вещественные, символьные, логические. Запись целых чисел в двоичном и шестнадцатеричном виде. Значения целых переменных как элементы кольца вычетов  $Z_2^{32}$ . Массивы. Строки и их представление в виде массивов символов.

Различные примеры алгоритмов: вычисление функций на последовательностях (записанных в массиве): сумма, минимум и максимум, минус бесконечность как максимум пустой последовательности, вычисление значения многочлена (схема Горнера).

Построение цикла "пока" с помощью инварианта: общая схема, иллюстрация на примерах вычисления НОД, быстрого возведения в степень, расширенного алгоритма Евклида, приближенного вычисления логарифма без использования разложения в ряд.

### 2.1.2. Устройство компьютера

Устройство компьютера: Фон-Неймановская архитектура, процессор, оперативная память и внешние устройства, хранение программы в памяти. Алгоритм работы процессора.

Устройство процессора: регистры процессора, типы команд. Общие и плавающие регистры. Выделенные регистры: счетчик команд PC (Program Counter), указатель стека SP (Stack Pointer), указатель кадра FP (Frame Pointer), регистр флагов CC (Condition Codes), общие регистры R0, R1, R2, ... Устройство конкретного процессора на примере Intel 80386.

Простейшие программы в кодах: на языке RTL виртуального процессора, на Ассемблере Intel 80386.

### 2.1.3. Основы языка Си

Введение в язык Си: базовые типы, массивы и указатели, функции, локальные и глобальные переменные. Управляющие конструкции языка Си. Примеры программ. Статические переменные и функции. Препроцессор: включение заголовочных файлов, задание констант с помощью "#define". Технология программирования: h-файлы и c-файлы, порядок описания и использования глобальных переменных и функций.

### 2.1.4. Приближенные вычисления

Работа с вещественными числами типа double в языке Си: представление целых и вещественных чисел, особенности приближенной арифметики (невыполнение законов ассоциативности и дистрибутивности). Программа, вычисляющая машинный эпсилон.

Вычисления с вещественными числами: суммирование рядов, вычисление стандартных функций (sin, cos, exp и т.п.) с помощью разложения в ряд. Численное интегрирование: формулы прямоугольников, трапеций, формула Симпсона.

Задача интерполяции. Интерполяционные многочлены Лагранжа и Ньютона. Сплайн-интерполяция.

Нахождение корня функции методом деления отрезка пополам и методом итераций Ньютона.

### 2.1.5. Стандартная библиотека Си: работа со строками, файлами, динамической памятью

Представление строк в языке Си и работа со строками. Стандартная библиотека: функции, описанные в стандартном заголовочном файле "string.h".

Работа с файлами в языке Си с использованием стандартной библиотеки ANSI (заголовочный файл "stdio.h").

Структуры языка Си и их использование.

Работа с динамической памятью в языке Си.

#### 2.1.6. Структуры данных, их использование и реализация

Структуры данных и их реализации: общее понятие структуры данных, описание и использование структуры данных на псевдокоде.

Стек как абстрактная структура данных и его реализация на базе массива. Проект "Стековый калькулятор" на языке Си. Аппаратный стек. Размещение локальных переменных в стеке и механизм передачи параметров функциям (роль стека и регистров SP, PC, FP). Рекурсия, примеры рекурсивных алгоритмов.

Обратная польская запись формул и программ. Стековый вычислитель. Примеры использования обратной польской записи: байткод языка Java, промежуточный язык PL (Intermediate Language), используемый в архитектуре Microsoft.NET. Язык Postscript и его использование, примеры.

Очередь как абстрактная структура данных и ее использование в программировании (очереди запросов к драйверу, очереди сообщений в оконной системе Windows и других оконных системах). Реализация очереди на базе циклического массива.

Линейный двунаправленный список как абстрактная структура данных. Однонаправленный список. Понятие ссылочной реализации структуры данных. Ссылочная реализация L2-списка.

Множество и нагруженное множество как абстрактные структуры данных. Битовая реализация ограниченного множества.

Наивная реализация множества и последовательный поиск. Непрерывная реализация множества, основанная на упорядочении его элементов и бинарном поиске. Построение и доказательство правильности программы бинарного поиска, основанное на схеме построения цикла с помощью инварианта.

### 2.1.7. Элементы теории структур данных

Задача сортировки массива. Оценка снизу числа сравнений в произвольном алгоритме сортировки ( $n \log n$ ), основанная на представлении алгоритма в виде бинарного дерева. Алгоритмы с оптимальным временем работы: пирамидальная сортировка HeapSort, сортировка слиянием. Быстрая (но не оптимальная!) сортировка QuickSort.

Реализация множества и нагруженного множества с помощью хеширования: идея хеширования, коллизии. Примеры хеш-функций. Способы разрешения коллизий: хеш-таблица как массив голов однонаправленных списков, хеш-таблица как обычный массив плюс массивы "мощность слоя" и "остаток".

Деревья, ссылочная реализация дерева. Алгоритм рекурсивного обхода дерева, примеры (файловая система и т.п.). Сбалансированное бинарное дерево, бинарный поиск по дереву, идея реализации множества с помощью сбалансированного дерева.

## 2.2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И ПРОГРАММИРОВАНИЕ В ОКОННЫХ СРЕДАХ

*Цели:* Овладение основными понятиями структурного программирования (класс, члены и методы классов, наследование и виртуальные методы), умение применять их в практических задачах. Понимание различий между традиционными языками (Си, Фортран, Паскаль), промежуточным (C++) и настоящими объектно-ориентированными языками (Java, C#, Visual Basic и т.п.) в организации работы с памятью и способах выполнения программ (компиляция, интерпретация, компиляция "на лету", сборка мусора). Умение писать простые программы на языках C++ и C# в системе программирования Microsoft.NET. Умение применять объектно ориентированный подход при программировании в оконных средах. Овладение технологией разработки оконных приложений в среде Microsoft.NET на языке C#. Умение применять параллельные подпроцессы (нити) для программирования оконных приложений и владение современными приемами программирования, основанными на использовании исключений ("Exception") для обработки ошибочных ситуаций, а также событий "Event" и делегатов языка C#.

2.2.1. Возникновение понятия класса (исполнителя, информационно-прочного модуля) как набора данных (членов класса) плюс набора программ, работающих над этими данными (методов класса). Язык C++, простейшие примеры классов.

- 2.2.2. Описание класса, разделение в С++ описания класса на интерфейс (h-файл) и реализацию (cpp-файл). Инлайновая реализация несложных методов.
- 2.2.3. Понятие объекта класса и различие между описанием класса и объектом класса (чертеж автомобиля и автомобиль). Конструктор и деструктор класса. Два способа создания объектов класса в языке С++: внутри переменной (в статической или стековой памяти), в динамической памяти с помощью оператора "new". Удаление объектов с помощью оператора "delete".
- 2.2.4. Ссылки (псевдонимы объектов) как новый тип данных языка С++, отсутствовавший в С. Важность модификатора const в языке С++. Константные указатели, ссылки и методы. Правильное описание прототипов методов как ключевой момент в освоении языка С++. Определение стандартных операторов языка С для классов на примере класса "Вектор на плоскости". Десять способов неправильного описания прототипов метода сложения векторов и единственный правильный способ.
- 2.2.5. Наследование и виртуальные методы. Программирование в оконных системах на примере библиотеки Framework в архитектуре Microsoft.NET. Обработка оконных сообщений и виртуальные методы классов типа "Окно". Перекрытие виртуальных методов как основа программирования графических приложений. Простейшие примеры графических программ на С++ в архитектуре Microsoft.NET.
- 2.2.6. Управляющие элементы окон (кнопки, текстовые поля, списки и т.д.), создание оконных форм. Программирование оконных приложений,

использующих управляющие элементы. Проект "Стековый калькулятор" на языке C++ в архитектуре Microsoft.NET.

- 2.2.7. Язык C# как "настоящий" объектно-ориентированный язык. Различие между традиционными и объектно-ориентированными языками по способам размещения объектов в памяти и доступа к ним. Три вида памяти в традиционных языках: статическая, стековая, динамическая. Управляемая динамическая память в объектно-ориентированных языках. Различие между переменной в C++ (содержит сам объект) и C# (содержат handle объекта, сам объект всегда расположен в управляемой динамической памяти). C++ как промежуточный язык между традиционными и объектно-ориентированными.
- 2.2.8. Управление объектами и динамической памятью в объектно-ориентированных языках. Сборщик мусора как ключевой элемент исполняющей системы любого объектно-ориентированного языка. Отличие компилируемых (традиционных) и интерпретируемых (объектно-ориентированных) языков. Компиляция "на лету" (Just In Time compiling) как альтернатива интерпретации.
- 2.2.9. Примеры программ на языке C#. Простейшая консольная программа – печать текущего времени. Свойства (properties) в языке C# как воображаемые переменные, у которых есть 2 метода "прочитать значение" get и "присвоить значение" set.
- 2.2.10. Простейшая графическая программа на языке C#: рисование графика функции.

- 2.2.11. Исключения "Exception" и их использование в программировании. Блоки try, catch и finally. Преимущества использования механизма исключений вместо анализа кодов возврата функций (кодов ошибок).
- 2.2.12. Использование нитей "Thread" для параллельного программирования. Почему без использования нитей невозможно программировать в оконных средах. Делегаты C# как способ передачи указателей на методы классов. Класс Thread как пульт управления нитью. Создание и запуск нити. Пример: графическая программа "Часы" на языке C#.
- 2.2.13. Синхронизация нитей: исключение совместного доступа к критическим объектам при помощи мониторов (класс Monitor) или с помощью ключевого слова "lock". Примеры.
- 2.2.14. Сигнализация между нитями с помощью объектов типа AutoResetEvent и ManualResetEvent, примеры.
- 2.2.15. Создание оконных форм с управляющими элементами в языке C# с использованием среды разработки Microsoft.NET. Проект "Стековый калькулятор" на языке C#.

### 3. ФОРМАТЫ ДАННЫХ И ЯЗЫКИ РАЗМЕТКИ

*Цели: Понимание методов представления и описания слабоструктурированных данных. Знание основных элементов языков разметки. Умение создавать простейшие электронные публикации. Умение использовать слабоструктурированные данные для прикладных программ и офисных приложений.*

- 3.1. Язык HTML описания гипертекстов в Internet. Основные принципы: структура файла, теги, открывающие и закрывающие теги, атрибуты тегов. Заголовок <head> и тело <body> страницы. Наиболее важные теги: заголовки <h1>, <h2>, ..., абзацы <p>, гипертекстовые ссылки <a href="...">, якоря в тексте <a name="...">, добавление картинок  нумерованные и нумерованные списки <ul> и <ol>.
- 3.2. Более сложные возможности HTML: вставка переформатированного текста; вставка специальных символов; задание таблиц. Использование Java-апплетов и Java-скриптов. Использование фреймов. Передача данных от клиента к HTTP-серверу с помощью форм и формирование страниц "на лету" сервером. Описание стилей страниц с использованием языка CSS (Cascading Style Sheet), примеры.
- 3.3. Язык описания XML. Сходство с HTML и основные отличия: строгость (обязательность закрывающих тегов, обязательность корневого тега), ориентация на описания содержания, а не на визуальное представление данных. Универсальность языка XML и примеры его использования. Схемы и их использование в XML.
- 3.4. TeX как язык для подготовки электронных публикаций. Форматы, Стили. Процесс компиляции, просмотра и печати документа.

## 4. КОМПЬЮТЕРНЫЕ СЕТИ

*Цели: Понимание принципов построения сетей передачи данных (передача пакетов вместо коммутации каналов) и знание классификации сетей по размеру, роли отдельных узлов и используемым протоколам. Понимание ключевого понятия протокола и уровневой модели описания и реализации протоколов. Знание основных протоколов физического и канального уровня (среда передачи, Манчестерское кодирование битов и байтов, случайные и детерминированные алгоритмы получения доступа к среде). Знание основных протоколов сети Internet и места их в уровневой модели классификации протоколов, понимание, почему протоколы Internet не зависят от протоколов нижних уровней (физического, канального, сетевого) и от среды передачи. Знание системы адресов Internet (IP-номер и порт, параллельная система символических имен). Понимание предназначения и различий базовых протоколов Internet (IP, UDP, TCP) и предназначения основных протоколов прикладного уровня (Telnet, FTP, SMTP, HTTP, POP3, SSH). Понимание способа реализации IP-протокола с помощью модема и протокола PPP. Умение писать простейшие сетевые программы на C++ и C# с использованием интерфейса сокетов и с использованием классов, реализующих протоколы прикладных уровней (HTTP и т.п.) в языке C#.*

- 4.1. Типы сетей передачи информации: сети с коммутацией каналов и с коммутацией (с передачей) пакетов. Отличие компьютерных сетей как сетей с передачей пакетов от телефонных сетей как сетей с коммутацией каналов.
- 4.2. Классификация сетей: по размеру – локальные, городские и внутрикорпоративные, всемирные; по типу передачи – с коммутацией каналов и с передачей пакетов; по роли отдельных узлов сети – одноранговые сети и сети типа клиент-сервер; по используемым сетевым протоколам.
- 4.3. Центральное понятие в теории сетей – понятие протокола как систематизированного описания соглашений о взаимодействиях. Уровневое описание протоколов, почему оно необходимо. Модель ISO/OSI уровневого описания протоколов компьютерных сетей (International Standard Organization / Open System Interconnection). Примеры протоколов физического, канального, сетевого, транспортного и сессионного уровней. Основные протоколы физического и канального уровней: Манчестерское кодирование битов, случайные и детерминированные алгоритмы получения доступа к среде на примерах сетей Ethernet и Token Ring. Важнейшее различие между протоколами транспортного и сессионного уровней: передача дейтаграмм и установка виртуального соединения. Способы реализации протокола сессионного уровня над транспортным уровнем: передача контрольной суммы пакетов, повтор передачи искаженных пакетов, протокол скользящих окон для упреждающей передачи и исправления ошибок. Протоколы Internet: базовый протокол IP как пример протокола транспортного уровня, протокол TCP как пример протокола сессионного уровня. Независимость протоколов Internet от протоколов нижних уровней (физического, канального и сетевого) и, следовательно, от среды передачи.

- 4.4. Протоколы физического уровня: кодирование битов и байтов электрическими, радио и оптическими сигналами. Среда передачи, разновидности (коаксиальный кабель и витая пара, оптоволоконный кабель и т.п.). Синхронная и асинхронная передача. Понятие несущей частоты и теорема Найквиста. Манчестерское кодирование как пример синхронной передачи с автоподстройкой частоты.
- 4.5. Алгоритмы доступа к среде передачи в протоколах физического уровня. Протоколы со случайным и детерминированным доступом. Примеры детерминированных протоколов: маркерное кольцо и шина (Token Ring и Token Bus), двойная шина с распределенной очередью (DQDB – Distributed Queue Dual Bus), FDDI (Fiber Optic Distributed Dual Interface). Протокол со случайным доступом: Ethernet (CSMA/CD Carrier – Sense Multiple Access / Collision Detection). Описание алгоритма доступа к среде в протоколе Ethernet. Объяснение, почему не очень удачный протокол Ethernet стал наиболее распространенным. Отсутствие всякой защиты данных в сети Ethernet как сети широковещательного типа.
- 4.6. Форматы пакетов в протоколе Ethernet, система адресов в Ethernet, простые и широковещательные пакеты. Типы носителей в Ethernet (толстый и тонкий коаксиальный кабель и витая пара), способы подключения узлов и топология сетей Ethernet. Физические ограничения по размеру сети (длина сегмента 200 м, максимальное расстояние между узлами в разных сегментах не более 1500 м). Способы соединения сегментов сети: Хабы (Hub), повторители (Repeater), переключатели (Switch), шлюзы (Gateway). В чем принципиальные различия между этими устройствами.

- 4.7. Башня TCP/IP протоколов сети Internet. Протокол IP как базовый протокол в Internet. Уровни протоколов IP (транспортный, дейтаграммного типа) и TCP (сессионный, основанный на установке виртуального соединения и гарантированной передачи сообщений произвольной длины). Система адресов в Internet: IP-адреса, разделение IP адреса на номер сети (network) и номер узла (host). Сетевая маска как способ выделения номера сети. Сети классов A, B, C. Организация сегментов сети: сетевая маска и номер сети одинаковы во всех узлах сегмента. Статическое и динамическое распределение IP-адресов, протокол DHCP динамического распределения IP-адресов.
- 4.8. Полный адрес процесса в Internet: IP-номер как адрес компьютера и порт как номер процесса внутри компьютера. Сокет как конечная точка взаимодействия и пара (IP-номер, порт) как адрес сокета. Протокол TCP: различие при установке соединения между сервером (пассивная сторона) и клиентом (активная сторона). Отсутствие различий между сервером и клиентом после установки соединения. Известные сервисы и выделенные порты. Привилегированные и непривилегированные порты и динамическое выделение портов для клиентов.
- 4.9. Параллельная система символических адресов в Internet. Иерархическая система адресов, области. Система серверов имен (name-servers) в Internet. Понятие зоны ответственности сервера, первичные (primary) и вторичные (secondary) серверы имен. Отображение символических имен на IP-адреса и обратное отображение с помощью зоны "in-addr.arpa".
- 4.10. Модем (модулятор-демодулятор) как устройство для передачи цифровых данных по аналоговой линии. Недостатки модема и перспективы развития телефонных сетей. Устройство и протоколы Hayes-совместимого

модема: режим команд и режим данных, AT-команды, примеры (команды набора номера, инициализации модема, установка различных параметров, разрыв связи, переход из режима данных в режим команд и т.д.). Программа "Терминал" и работа на удаленном компьютере через модем. Временное подключение к Internet с помощью модема: протокол PPP как реализация IP-протокола для телефонных линий. Протокол UUCP и два способа работы с электронной почтой: 1) с помощью протокола UUCP (без подключения к Internet); 2) с помощью временного подключения к Internet через протокол PPP и с использованием протокола работы с удаленным почтовым ящиком POP3.

4.11. Наиболее важные протоколы прикладных уровней в сети Internet, построенные над протоколом TCP как базовым: передача файлов FTP (порт 21), электронная почта SMTP (25), протокол работы с удаленным почтовым ящиком POP3 (110), протокол удаленного терминала Telnet (23), протокол удаленного терминала с защитой данных SSH (22), протокол передачи гипертекста "Всемирной паутины" HTTP (80). Общие принципы построения всех протоколов: серверы, или Internet-демоны, понимают обычные текстовые команды на английском языке. Наборы команд для протокола электронной почты SMTP и протокола передачи гипертекста HTTP. Иллюстрация общения с Internet-демоном с помощью программы "Telnet" на примерах почтового демона SMTP (порт 25) и демона всемирной паутины HTTP (порт 80).

4.12. Интерфейс сокетов как интерфейс для программирования сетевого обмена на примере программирования в сети Internet. Различия между клиентом и сервером при установке соединения. Установка соединения клиентом: создание сокета,

преобразование символического адреса в IP-адрес, заполнение адреса сервера (IP-номер и порт), выполнение команды "connect". Установка соединения сервером: создание сокета, заполнения номера порта в адресе сокета, выполнение команды "listen". Принятие входящего соединения с помощью команды "accept" и создание нового сокета для передачи данных. Полная симметрия между сервером и клиентом после установки соединения. Передача данных: аналогия с записью и чтением в случае файлов. Понятие потока данных и общий интерфейс для работы с файлами и с сетями. Примеры программ, использующих интерфейс сокетов, на языках C, C++ и C#. Классы в C#, поддерживающие сетевой обмен по протоколу HTTP, примеры программ.

## 5. УПРАВЛЕНИЕ ИНФОРМАЦИЕЙ

*Цели: Знание современных методов представления и обработки информации. Формирование навыков и умений разработки баз данных. Владение языками запросов. Знание принципов и методик создания информационных систем различного назначения. Представление о задачах искусственного интеллекта, используемых для их решения алгоритмах. Знание методов разработки баз знаний, экспертных систем, обработки информации в глобальной сети.*

- 5.1. Понятия информационной системы, базы данных. Концепции технологии доступа к данным: модели данных с навигацией, реляционная модель, нелинейная модель. Инфологическое проектирование. Построение ER-моделей. Язык UML: основные понятия.
- 5.2. Реляционная теория: отношения, реляционная алгебра, теория нормализации, реляционное исчисление, целостность. Определение реляционной модели. Преобразование ER-моделей в структуру баз данных.
- 5.3. Структурированный язык запросов SQL: история, структура, стандарты. Язык определения данных. Формирование запроса, базовая функциональность. Операции обновления. Понятие транзакции. Представления. Курсор.
- 5.4. Концепции объектно-ориентированных баз данных. Гибридные и расширенные реляционные формы управления объектами. Структурированные и слабоструктурированные данные: способы описания,

использование языка XML для описания данных, смешанные языки запросов.

- 5.5. Обработка информации и сетевые архитектуры. Многоуровневая технология. Теория распределенных систем К.Дейта. Архитектуры аппаратных платформ и реляционные базы данных. Параллельное выполнение операций и их реализация. Фрагментация и репликация отношений.
- 5.6. Введение в интеллектуальные системы. История развития и области приложения искусственного интеллекта. Базы знаний. Экспертные системы. Автоматическое доказательство. Понимание естественных языков. Машинное обучение.
- 5.7. Модели представления знаний: продукционная модель, семантические сети, фреймы. Вывод на знаниях. Прямой и обратный вывод. Стратегии поиска. Нечеткие знания.
- 5.8. Представление данных и знаний в глобальной сети. Онтологии и онтологические системы. Методологии создания онтологий. Программные агенты и мультиагентные системы. Информационный поиск в Глобальной сети.

## 6. КОМПЬЮТЕРНАЯ ГРАФИКА

*Цели:* Знание методов представления изображений в компьютере (видеопамять, растеризация, представление пикселей и видеорежимы). Знание основных алгоритмов двумерной графики (растеризация линий, сглаживание, закрашивание областей). Умение писать простейшие графические программы в среде разработки Microsoft.NET. Понимание основного принципа трехмерной графики: использование отдельного процессора для вычисления трехмерных изображений и передача описаний трехмерных объектов в видеопроцессор вместо непосредственного рисования. Знание основных алгоритмов трехмерной графики (алгоритмы Гуро и Фонга закрашивания треугольника, использование нормалей, использование буфера глубины для удаления невидимых линий). Знание основ библиотеки OpenGL (система координат, способы описания объектов, преобразования MODELVIEW и PROJECTION) и умение программировать простые трехмерные задачи с помощью OpenGL. Знакомство с языком VRML как примером языка описания трехмерных миров.

- 6.1. Представление изображений в компьютере: видеорежимы, видеопамять, растеризация изображений. Двумерная графика: алгоритмы рисования линий Брезенхема (на примерах прямой линии, окружности и эллипса). Алгоритм сглаживания линий (antialiasing). Рекурсивный алгоритм закрашивания области. Векторная графика и ее отличие от растровой, способы задания масштабируемых объектов. Примеры простых графических программ на языке C# в среде разработки Microsoft.NET.

- 6.2. Основы трехмерной графики. Использование отдельного процессора, встроенного в конструкцию видеокарты, и передача от компьютера в видеокарту описаний трехмерных объектов вместо построения изображения в видеопамяти. Алгоритмы построения трехмерных изображений: удаление невидимых линий с помощью буфера глубины (Z-буфера). Освещение граней, использование нормалей. Модели источников света. Алгоритмы Гуро и Фонга закрашивания треугольника или выпуклого многоугольника в пространстве.
- 6.3. Библиотека OpenGL для поддержки трехмерной графики. Основная идея: описание трехмерного мира ("Science Graph") вместо построения изображения. Проективные координаты. Матрицы "MODELVIEW" и "PROJECTION". Типы проекций на двумерную плоскость: ортогональная и проективная (центральная) проекции. Параметры центральной проекции: угол зрения, ближайшая и удаленная плоскости отсечения. Представление трехмерных объектов в виде разбиения на треугольники или выпуклые многоугольники. Использование нормалей для сглаживания изображения поверхностей, аппроксимированных многоугольниками. Описания источников света. Задание текстур.
- 6.4. Язык VRML описания трехмерных объектов и трехмерных миров. Основы языка, простейшие примеры. Описание простейших геометрических объектов (ящик, сфера, цилиндр) и геометрических объектов произвольной формы, представленных в виде разбиения на плоские грани. Использование нормалей и другие способы сглаживания поверхностей. Определение источников света. Анимация: передача сообщений от одних узлов другим; узлы, обеспечивающие анимацию (сенсоры и интерполяторы), примеры.