Дисциплина: **Информационные системы и технологии**

специальность: **информатики**

курс, группа: **1 к 1,2 г**

форма обучения: **заочная**

период проведения занятий: **01 июня 2020 г.**

вид занятий, кол-во часов: Консультация, 4 часа

преподаватель: **Астахов В.К.**

электронная почта преподавателя: **vadast@mail.ru**

**Литература:**

## 1 Основная литература:

1. Балдин К.В. Информационные системы в экономике [Электронный ресурс]: учебник/ Балдин К.В., Уткин В.Б.— Электрон. текстовые данные. — М.: Дашков и К, 2015. — 395 c.— Режим доступа: http://www.iprbookshop.ru/24785.— ЭБС «IPRbooks», по паролю (Гриф УМО)
2. Информационные системы и технологии в экономике и управлении. Экономические информационные системы: учебное пособие / Е. В. Акимова, Д. А. Акимов, Е. В. Катунцов, А. Б. Маховиков. — Саратов: Вузовское образование, 2016. — 172 c. — ISBN 2227-8397. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/47675.html (дата обращения: 19.11.2019). — Режим доступа: для авторизир. пользователей
3. Уткин, В. Б. Информационные системы и технологии в экономике: учебник для вузов / В. Б. Уткин, К. В. Балдин. — М.: ЮНИТИ-ДАНА, 2017. — 336 c. — ISBN 5-238-00577-6. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/71196.html (дата обращения: 19.11.2019). — Режим доступа: для авторизир. пользователей (гриф МО)

## 2. Дополнительная литература:

1. Косиненко, Н. С. Информационные системы и технологии в экономике [Электронный ресурс]: учебное пособие / Н. С. Косиненко, И. Г. Фризен. — Электрон. текстовые данные. — М.: Дашков и К, Ай Пи Эр Медиа, 2017. — 304 c. — 978-5-394-01730-8. — Режим доступа: http://www.iprbookshop.ru/57134.html.
2. Мельников, А. В. Информационные системы в бухгалтерском учете (теория и практика): учебное пособие / А. В. Мельников, С. Н. Черняева; под редакцией Л. А. Коробова. — Воронеж: Воронежский государственный университет инженерных технологий, 2015. — 78 c. — ISBN 978-5-00032-107-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/50631.html (дата обращения: 19.11.2019). — Режим доступа: для авторизир. пользователей
3. Стешин, А. И. Информационные системы в организации [Электронный ресурс]: учебное пособие / А. И. Стешин. — 2-е изд. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2019. — 194 c. — 978-5-4487-0385-0. — Режим доступа: http://www.iprbookshop.ru/79629.html

**Интернет- ресурсы**

1. Бесплатные видеокурсы и видеолекции онлайн IT: Информационные системы: ERP, CRM ... Технологии открытых систем… [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www. edu.jobsmarket.ru ›
2. Бесплатные видеокурсы и видеолекции онлайн IT: Информационные системы: ERP, CRM ... Технологии открытых систем… [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www. edu.jobsmarket.ru ›
3. Бесплатные видеолекции: …темы: 2. Классификация информационных систем [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www. edu.jobsmarket.ru ›
4. Библиотека публикаций и форумы по разработке и применению информационных систем [Электронный ресурс]: раздел Информационные технологии. — Электрон. дан. — Режим доступа: http://www. sql.ru
5. Библиотека публикаций по применению систем управления бизнес–процессами [Электронный ресурс]: раздел Информационные технологии. — Электрон. дан. — Режим доступа: http://www. bpms.ru
6. Видеолекции: Информационные технологии... [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www. distanz.ru
7. Информационные технологии... видеолекци по предмету «Информационные технологии»… [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www.www.distanz.ru/
8. Информационные технологии... видеолекци по предмету «Информационные технологии»… [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www.www.distanz.ru/
9. Информационные системы и технологии. Практикум. [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www.edu.dvgups.ru/
10. Информационные технологии в экономике: Учеб. пособие. [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа:http://www.ecsocman.hse.ru/
11. Красильникова М.В. Информационные системы и технологии. Учебное пособие. [Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www.antigtu.ru/141
12. Курс лекций. В курсе рассматриваются принципы создания информационных технологий и систем, их внедрения и использования...[Электронный ресурс]: офиц.сайт. — Электрон. дан. — Режим доступа: http://www.edumarket.ru/library/itsystem/15330/
13. Научная электронная библиотека eLIBRARY.ru [Электронный ресурс]: раздел Информационные технологии. — Электрон. дан. — Режим доступа: http://www.elibrary.ru/ defaultx.asp
14. Научная электронная библиотека IPRbooks.ru [Электронный ресурс]: раздел Информационные технологии. — Электрон. дан. — Режим доступа: http://www. iprbooks.ru (по паролю)
15. Научная электронная онлайн-библиотека Порталус [Электронный ресурс]: раздел Информационные технологии. — Электрон. дан. — Режим доступа: http://www. portalus.ru

**Задания по темам и датам**

| **дата, время**  **занятия** | **вид, тема**  **занятия** | **кол-во часов** | **вопросы для изучения и обсуждения** | **контрольные вопросы, задания** |
| --- | --- | --- | --- | --- |
| 01.06.2020  14.05-15.35 | Консультация.  Темы 5,6 дисциплины | 2 | 1. Информационный процесс обмена данными. Сетевые информационные технологии.  2.Технологии поиска информации. СПС Консультант+, Гарант. | 1.Изучить представленный ниже материал и рекомендованную литературу.  (присылать материалы преподавателю для проверки НЕ НУЖНО) |
| 01.06.2020  15.35-17.15 | Консультация.  Темы 7,8 дисциплины | 2 | 3.Технология управления проектами на примере Project expert и Microsoft Project  4. Технология комплексной автоматизации деятельности предприятия на примере 1С:Предприятие 8.3 | 1.Изучить представленный ниже материал и рекомендованную литературу.  2.Выполнить задания 1-6 (задачи 1-8), представленные ниже.  (присылать материалы преподавателю для проверки НЕ НУЖНО) |

**Рекомендации и требования к выполнению указанных заданий**

Краткий материал по дисциплине **Информационные системы и технологии** для консультаций на 01.06.2020 г.

***Консультация №1***

**1. Информационный процесс обмена данными. Сетевые информационные технологии**

**Обзор и архитектура вычислительных сетей**

**Основные определения и термины**

*Сеть* – это совокупность объектов, образуемых устройствами передачи и обработки данных. Международная организация по стандартизации определила вычислительную сеть *как последовательную бит-ориентированную передачу информации между связанными друг с другом независимыми устройствами.*

Сети обычно находится в частном ведении пользователя и занимают некоторую территорию и по территориальному признаку разделяются на:

Локальные вычислительные сети (ЛВС) или Local Area Network (LAN), расположенные в одном или нескольких близко расположенных зданиях. ЛВС обычно размещаются в рамках какой-либо организации (корпорации, учреждения), поэтому их называют корпоративными.

Распределенные компьютерные сети, глобальные или Wide Area Network (WAN), расположенные в разных зданиях, городах и странах, которые бывают территориальными, смешанными и глобальными. В зависимости от этого глобальные сети бывают четырех основных видов: городские, региональные, национальные и транснациональные. В качестве примеров распределенных сетей очень большого масштаба можно назвать: Internet, EUNET, Relcom, FIDO.

В состав сети в общем случае включается следующие элементы:

сетевые компьютеры (оснащенные сетевым адаптером);

каналы связи (кабельные, спутниковые, телефонные, цифровые, волоконно-оптические, радиоканалы и др.);

различного рода преобразователи сигналов;

сетевое оборудование.

Различают два понятия сети: *коммуникационная сеть* и *информационная сеть* (рис. 1.1).

*Коммуникационная сеть* предназначена для передачи данных, также она выполняет задачи, связанные с преобразованием данных. Коммуникационные сети различаются по типу используемых физических средств соединения.

*Информационная сеть* предназначена для хранения информации и состоит из *информационных систем*. На базе коммуникационной сети может быть построена группа информационных сетей:

Под *информационной системой* следует понимать систему, которая является поставщиком или потребителем информации.



Рис. 1.1 Информационные и коммуникационные сети

Компьютерная сеть состоит из информационных систем и каналов связи.

Под *информационной системой* следует понимать объект, способный осуществлять хранение, обработку или передачу информация. В состав *информационной системы* входят: компьютеры, программы, пользователи и другие составляющие, предназначенные для процесса обработки и передачи данных. В дальнейшем информационная система, предназначенная для решения задач пользователя, будет называться – *рабочая станция (client)*. Рабочая станция в сети отличается от обычного персонального компьютера (ПК) наличием *сетевой карты* (*сетевого адаптера*), канала для передачи данных и сетевого программного обеспечения.

Под *каналом связи* следует понимать путь или средство, по которому передаются сигналы. Средство передачи сигналов называют *абонентским,* или *физическим, каналом*.

*Каналы связи (data link)* создаются по линиям связи при помощи сетевого оборудования и физических средств связи. Физические средства связи построены на основе витых пар, коаксиальных кабелей, оптических каналов или эфира. Между взаимодействующими информационными системами через физические каналы коммуникационной сети и узлы коммутации устанавливаются *логические каналы.*

*Логический канал* – это путь для передачи данных от одной системы к другой. Логический канал прокладывается по маршруту в одном или нескольких физических каналах. *Логический канал* можно охарактеризовать, как маршрут, проложенный через физические каналы и узлы коммутации.

Информация в сети передается *блоками данных* по процедурам обмена между объектами. Эти процедуры называют *протоколами передачи данных.*

*Протокол –* это совокупность правил, устанавливающих формат и процедуры обмена информацией между двумя или несколькими устройствами.

Загрузка сети характеризуется параметром, называемым *трафиком*. *Трафик (traffic) –* это поток сообщений в сети передачи данных. Под ним понимают количественное измерение в выбранных точках сети числа проходящих *блоков данных* и их длины, выраженное в битах в секунду.

Существенное влияние на характеристику сети оказывает *метод доступа*. *Метод доступа* – это способ определения того, какая из рабочих станций сможет следующей использовать канал связи и как управлять доступом к каналу связи (кабелю).

В сети все рабочие станции физически соединены между собою каналами связи по определенной структуре, называемой *топологией*. *Топология* – это описание физических соединений в сети, указывающее какие рабочие станции могут связываться между собой. Тип топологии определяет производительность, работоспособность и надежность эксплуатации рабочих станций, а также время обращения к файловому серверу. В зависимости от топологии сети используется тот или иной метод доступа.

Состав основных элементов в сети зависит от ее архитектуры. *Архитектура* – это концепция, определяющая взаимосвязь, структуру и функции взаимодействия рабочих станций в сети. Она предусматривает логическую, функциональную и физическую организацию технических и программных средств сети. Архитектура определяет принципы построения и функционирования аппаратного и программного обеспечения элементов сети.

В основном выделяют три вида архитектур: архитектура *терминал – главный компьютер*, архитектура *клиент – сервер* и *одноранговая* архитектура.

Современные сети можно классифицировать по различным признакам: по удаленности компьютеров, топологии, назначению, перечню предоставляемых услуг, принципам управления (централизованные и децентрализованные), методам коммутации, методам доступа, видам среды передачи, скоростям передачи данных и т. д. Все эти понятия будут рассмотрены более подробно при дальнейшем изучении курса.

**Преимущества использования сетей**

Компьютерные сети представляют собой вариант сотрудничества людей и компьютеров, обеспечивающего ускорение доставки и обработки информации. Объединять компьютеры в сети начали более 30 лет назад. Когда возможности компьютеров выросли и ПК стали доступны каждому, развитие сетей значительно ускорилось.

Соединенные в сеть компьютеры обмениваются информацией и совместно используют периферийное оборудование и устройства хранения информации рис. 1.2.

Per_ust

Рис. 1.1 Использование периферийного оборудования

С помощью сетей можно разделять ресурсы и информацию. Ниже перечислены основные задачи, которые решаются с помощью рабочей станции в сети, и которые трудно решить с помощью отдельного компьютера:

Компьютерная сеть позволит совместно использовать периферийные устройства, включая:

принтеры;

плоттеры;

дисковые накопители;

приводы CD-ROM;

дисководы;

стримеры;

сканеры;

факс-модемы;

Компьютерная сеть позволяет совместно использовать информационные ресурсы:

каталоги;

файлы;

прикладные программы;

игры;

базы данных;

текстовые процессоры.

Компьютерная сеть позволяет работать с многопользовательскими программами, обеспечивающими одновременный доступ всех пользователей к общим базам данных с блокировкой файлов и записей, обеспечивающей целостность данных. Любые программы, разработанные для стандартных ЛВС, можно использовать в других сетях.

Совместное использование ресурсов обеспечит существенную экономию средств и времени. Например, можно коллективно использовать один лазерный принтер вместо покупки принтера каждому сотруднику или беготни с дискетами к единственному принтеру при отсутствии сети.

Организация электронной почты. Можно использовать *ЛВС* как почтовую службу и рассылать служебные записки, доклады и сообщения другим пользователям.

**Архитектура сетей**

Архитектура сети определяет основные элементы сети, характеризует ее общую логическую организацию, техническое обеспечение, программное обеспечение, описывает методы кодирования. Архитектура также определяет принципы функционирования и интерфейс пользователя.

В данном курсе будет рассмотрено три вида архитектур:

архитектура терминал – главный компьютер;

одноранговая архитектура;

архитектура клиент – сервер.

**Архитектура терминал – главный компьютер**

Архитектура терминал – главный компьютер (terminal – host computer architecture) – это концепция информационной сети, в которой вся обработка данных осуществляется одним или группой главных компьютеров.

R3

Рис. 1.1 Архитектура терминал – главный компьютер

Рассматриваемая архитектура предполагает два типа оборудования:

Главный компьютер, где осуществляется управление сетью, хранение и обработка данных.

Терминалы, предназначенные для передачи главному компьютеру команд на организацию сеансов и выполнения заданий, ввода данных для выполнения заданий и получения результатов.

*Главный компьютер* через мультиплексоры передачи данных (МПД) взаимодействуют с терминалами, как представлено на рис. 1.3.

Классический пример архитектуры сети с главными компьютерами – системная сетевая архитектура (System Network Architecture – SNA).

**Одноранговая архитектура**

Одноранговая архитектура (peer-to-peer architecture) – это концепция информационной сети, в которой ее ресурсы рассредоточены по всем системам. Данная архитектура характеризуется тем, что в ней все системы равноправны.

К *одноранговым* сетям относятся малые сети, где любая рабочая станция может выполнять одновременно функции файлового сервера и рабочей станции. В *одноранговых* *ЛВС* дисковое пространство и файлы на любом компьютере могут быть общими. Чтобы ресурс стал общим, его необходимо отдать в общее пользование, используя службы удаленного доступа сетевых одноранговых операционных систем. В зависимости от того, как будет установлена защита данных, другие пользователи смогут пользоваться файлами сразу же после их создания. *Одноранговые* *ЛВС* достаточно хороши только для небольших рабочих групп.



Рис. 1.1 Одноранговая архитектура

*Одноранговые* *ЛВС* являются наиболее легким и дешевым типом сетей для установки. Они на компьютере требуют, кроме сетевой карты и сетевого носителя, только операционной системы *Windows* 95 или *Windows for Workgroups.* При соединении компьютеров, пользователи могут предоставлять ресурсы и информацию в совместное пользование.

Одноранговые сети имеют следующие преимущества:

они легки в установке и настройке;

отдельные ПК не зависят от выделенного сервера;

пользователи в состоянии контролировать свои ресурсы;

малая стоимость и легкая эксплуатация;

минимум оборудования и программного обеспечения;

нет необходимости в администраторе;

хорошо подходят для сетей с количеством пользователей, не превышающим десяти.

Проблемой одноранговой архитектуры является ситуация, когда компьютеры отключаются от сети. В этих случаях из сети исчезают виды *сервиса*, которые они предоставляли. Сетевую безопасность одновременно можно применить только к одному ресурсу, и пользователь должен помнить столько паролей, сколько сетевых ресурсов. При получении доступа к разделяемому ресурсу ощущается падение производительности компьютера. Существенным недостатком одноранговых сетей является отсутствие централизованного администрирования.

Использование одноранговой архитектуры не исключает применения в той же сети также архитектуры «терминал – главный компьютер» или архитектуры «клиент – сервер».

**Архитектура клиент – сервер**

*Архитектура клиент – сервер* (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов (рис. 1.5). Рассматриваемая архитектура определяет два типа компонентов: *серверы и клиенты*.

*Сервер -* это объект, предоставляющий *сервис* другим объектам сети по их запросам. *Сервис* – это процесс обслуживания клиентов.



Рис. 1.1 Архитектура клиент – сервер

Сервер работает по заданиям клиентов и управляет выполнением их заданий. После выполнения каждого задания сервер посылает полученные результаты клиенту, пославшему это задание.

Сервисная функция в архитектуре клиент – сервер описывается комплексом прикладных программ, в соответствии с которым выполняются разнообразные прикладные процессы.

Процесс, который вызывает сервисную функцию с помощью определенных операций, называется *клиентом*. Им может быть программа или пользователь. На рис. 1.6 приведен перечень сервисов в архитектуре клиент – сервер.

*Клиенты* – это рабочие станции, которые используют ресурсы сервера и предоставляют удобные *интерфейсы пользователя*. *Интерфейсы пользователя* это процедуры взаимодействия пользователя с системой или сетью.

Клиент является инициатором и использует электронную почту или другие сервисы сервера. В этом процессе клиент запрашивает вид обслуживания, устанавливает сеанс, получает нужные ему результаты и сообщает об окончании работы.



Рис. 1.2 Модель клиент-сервер

В *сетях с выделенным файловым сервером* на выделенном автономном *ПК* устанавливается серверная сетевая операционная система. Этот *ПК* становится *сервером.* Программное обеспечение (*ПО*)*,* установленное на рабочей станции, позволяет ей обмениваться данными с сервером. Наиболее распространенные сетевые операционная системы:

NetWare фирмы Novel;

Windows NT фирмы Microsoft;

UNIX фирмы AT&T;

Linux.

Помимо сетевой операционной системы необходимы сетевые прикладные программы, реализующие преимущества, предоставляемые сетью.

Сети на базе серверовимеют лучшие характеристики и повышенную надежность. Сервервладеет главными ресурсами сети,к которым обращаются остальные рабочие станции.

В современной клиент – серверной архитектуре выделяется четыре группы объектов: клиенты, серверы, данные и сетевые службы. Клиенты располагаются в системах на рабочих местах пользователей. Данные в основном хранятся в серверах. Сетевые службы являются совместно используемыми серверами и данными. Кроме того службы управляют процедурами обработки данных.

Сети клиент – серверной архитектуры имеют следующие преимущества:

позволяют организовывать сети с большим количеством рабочих станций;

обеспечивают централизованное управление учетными записями пользователей, безопасностью и доступом, что упрощает сетевое администрирование;

эффективный доступ к сетевым ресурсам;

пользователю нужен один пароль для входа в сеть и для получения доступа ко всем ресурсам, на которые распространяются права пользователя.

Наряду с преимуществами сети клиент – серверной архитектуры имеют и ряд недостатков:

неисправность сервера может сделать сеть неработоспособной, как минимум потерю сетевых ресурсов;

требуют квалифицированного персонала для администрирования;

имеют более высокую стоимость сетей и сетевого оборудования.

**Выбор архитектуры сети**

Выбор архитектуры сети зависит от назначения сети, количества рабочих станций и от выполняемых на ней действий.

Следует выбрать одноранговую сеть, если:

количество пользователей не превышает десяти;

все машины находятся близко друг от друга;

имеют место небольшие финансовые возможности;

нет необходимости в специализированном сервере, таком как сервер БД, факс-сервер или какой-либо другой;

нет возможности или необходимости в централизованном администрировании.

Следует выбрать клиент серверную сеть, если:

количество пользователей превышает десяти;

требуется централизованное управление, безопасность, управление ресурсами или резервное копирование;

необходим специализированный сервер;

нужен доступ к глобальной сети;

требуется разделять ресурсы на уровне пользователей.

**Вопросы**

1. Дать определение сети.
2. Чем отличается коммуникационная сеть от информационной сети?
3. Как разделяются сети по территориальному признаку?
4. Что такое информационная система?
5. Что такое каналы связи?
6. Дать определение физического канала связи.
7. Дать определение логического канала связи.
8. Как называется совокупность правил обмена информацией между двумя или несколькими устройствами?
9. Как называется объект, способный осуществлять хранение, обработку или передачу данных, в состав, которого входят компьютер, программное обеспечение, пользователи и др. составляющие, предназначенные для процесса обработки и передачи данных?
10. Каким параметром характеризуется загрузка сети?
11. Что такое метод доступа?
12. Что такое совокупность правил, устанавливающих процедуры и формат обмена информацией?
13. Чем отличается рабочая станция в сети от обычного персонального компьютера?
14. Какие элементы входят в состав сети?
15. Как называется описание физических соединений в сети?
16. Что такое архитектура сети?
17. Как назвать способ определения, какая из рабочих станций сможет следующей использовать канал связи?
18. Перечислить преимущества использования сетей.
19. Чем отличается одноранговая архитектура от клиент серверной архитектуры?
20. Каковы преимущества крупномасштабной сети с выделенным сервером?
21. Какие сервисы предоставляет клиент серверная архитектура?
22. Преимущества и недостатки архитектуры терминал – главный компьютер.
23. В каком случае используется одноранговая архитектура?
24. Что характерно для сетей с выделенным сервером?
25. Как называются рабочие станции, которые используют ресурсы сервера?
26. Что такое сервер?

**2.Технологии поиска информации. СПС Консультант+, Гарант**

**Информационно-поисковые системы**

Современный этап развития цивилизации характеризуется переходом наиболее развитой части человечества от индустриального общества к информационному. Одним из наиболее ярких явлений этого процесса является возникновение и развития глобальной информационной компьютерной сети. В данной работе рассматриваются теоретические основы информационного поиска, классификация и разновидности информационно поисковых систем. При появлении сети Интернет проблема поиска становилась более актуальной. Интернет - всемирная компьютерная сеть, представляющая собой единую информационную среду и позволяющая получить информацию в любое время. Но с другой стороны в Интернете храниться очень много полезной информации, но для поиска её требуется затрачивать много времени. Эта проблема послужила поводом к появлению поисковых машин. Информационные системы являются основным средством, инструментарием решения задач информационного обеспечения различных видов деятельности и наиболее бурно развивающейся отраслью индустрии информационных технологий. Структура информационной системы.

**Развитие поисковых систем**

Исторические предпосылки развития поисковых систем. Обратимся к истории возникновения сети Internet, которая была создана в связи с возникшей необходимостью совместного использования информационных ресурсов, распределенных между различными компьютерными системами. Большинство первых приложений, включая FTP и электронную почту, были разработаны исключительно для обмена данными между хост-компьютерами Internet. Другие приложения, такие как Telnet, создавались для того, чтобы пользователь получил возможность доступа не только к информации, но и к рабочим ресурсам удаленной системы. По мере развития Internet (увеличения пользователей и хост-компьютеров) прежние методы обмена данными перестали отвечать возросшим потребностям пользователей. Возникла необходимость разработки новых способов поиска сетевых ресурсов и доступа к ним, которые позволяли бы использовать информацию независимо от ее формата и расположения. Для удовлетворения таких потребностей сначала были созданы поисковая система Archie, решающая задачу локализации ресурсов на FTP-сервере, и система Gopher, упрощающая доступ к различным сетевым ресурсам. Затем были разработаны сетевые информационные системы WWW и WAIS, предлагающие абсолютно новые методы получения информации. Принципы работы этих систем позволяют легко ориентироваться в огромном количестве информационных ресурсов без необходимости предоставления механизмов работы самой сети Internet. Такой подход позволяет говорить уже не просто о ресурсах взаимосвязанных компьютерных систем, а об особых информационных пространствах сети. Система Archie представляет собой комплекс программных средств, работающих со специальными базами данных. В этих базах данных содержится постоянно пополняющаяся информация о файлах, к которым можно получить доступ через сервис FTP. Пользуясь услугами системы Archie, можно осуществить поиск файла по шаблону его имени. При этом пользователь получит список файлов с точным указанием места их хранения в сети, а также с информацией о типе, времени создания и размере файлов. Доступ к информационно-поисковой системе Archie может осуществляться различными путями, начиная от запросов по электронной почте и с помощью сервиса Telnet и заканчивая использованием графических Archie-клиентов. Система Gopher была разработана для упрощения процесса локализации FTP-ресурсов Internet и для более удобного представления сведений о содержании хранящихся на FTP-серверах файлов. Система Gopher дает возможность в удобной форме (в виде меню) представлять пользователям об имеющихся файлах и их содержании. Меню Gopher-серверов могут содержать ссылки на другие Gopher- и FTP-серверы. Таким образом, пользователь получает возможность “путешествовать” по Internet, не обращая внимания на местонахождение интересующих его ресурсов, и получать доступ к этим ресурсам. Под информационной системой в дальнейшем понимается - организованная совокупность программно - технических и других вспомогательных средств, технологических процессов и функционально - определённых групп работников, обеспечивающих сбор, представление и накопление информационных ресурсов в определённой предметной области, поиск и выдачу сведений, необходимых для удовлетворения информационных потребностей установленного контингента пользователей - абонентов системы.

**Особенности поисковых систем**

Особенности поисковых систем. В работе поисковый процесс представлен четырьмя стадиями: формулировка (происходит до начала поиска); действие (начинающийся поиск); обзор результатов (результат, который пользователь видит после поиска); и усовершенствование (после обзора результатов и перед возвращением к поиску с иной формулировкой той же потребности). Более удобная нелинейная схема поиска информации состоит из следующих этапов:

- фиксация информационной потребности на естественном языке;

- выбор нужных поисковых сервисов сети и точная формализация записи информационной потребности на конкретных информационно-поисковых языках (ИПЯ);

- выполнение созданных запросов;

- предварительная обработка и выборка полученных списков ссылок на документы;

- обращение по выбранным адресам за искомыми документами;

- предварительный просмотр содержимого найденных документов;

- сохранение релевантных документов для последующего изучения;

- извлечение из релевантных документов ссылок для расширения запроса;

- изучение всего массива сохраненных документов;

- если информационная потребность не полностью удовлетворена, то возврат к первому этапу.

**Структура поисковой системы**

**Архитектура поисковой системы**

Рассмотрим классическую архитектуру, которая чаще всего реализована на корпоративных сайтах и информационных порталах.

Существует клиентская вычислительная машина под управлением ОС Windows и существует Web-сервер под управлением UNIX-подобной ОС. На стороне клиента запущен типичный браузер, такой как Netscape. На стороне сервера запущен web сервер, который обслуживает запросы от браузера, передавая запросы презентационному слою понимающему CGI. Презентационный слой передает запросы к поисковому механизму в случае вызова услуги поиска или отображает наполнение (content) сайта. При работе администратора презентационный слой также может передавать запросы на инициализацию механизма индексации нового контента, который еще не индексирован. Это необходимо по той причине, что пока текст не индексирован, поиск в нем с помощью поисковой машины невозможен.

Идея заключается в следующем. Существуют мегабайты текстовой информации, и скорость поиска документов содержащих заданные ключевые слова отнимает очень многопроцессорного времени. Предположим, в 10 мегабайтах текстовой информации ключевое слово будет находиться в течение 10 секунд. И вот заходит посетитель на Ваш сайт, задает ключевые слова, вызывает услугу поиска и ждет 10 секунд, пока ваш сервер не выдаст ему результат. Предположим, случилось так, что одновременно запросило поиск 5 человек. Естественно, время ответа увеличится в 5 раз. Получается, что в среднем по 50 секунд пользователь будет ждать ответа от вашего сервера. Это не приемлемо, особенно если у Вас сотни мегабайт текстовой информации и время реакции системы будет катастрофически велико. Необходимо использовать другой подход при поиске ключевых слов в текстовой информации - время ответа сократить до миллисекунд.

**ER-модель поискового механизма**

Существует такая хорошая характеристика реляционных баз данных, как очень маленькое время выборки конкретной записи из миллионов других. Это достигается созданием, так называемого, индекса к таблице на какое-то из полей этой таблицы. Обычно индексы реализуются с применением алгоритма сбалансированного двоичного дерева. Предположим, у нас есть таблица, в которой всего один столбец и в каждой записи таблицы хранится фамилия человека. Предположим, мы загнали в такую таблицу 1 миллион фамилий. Нам необходимо проверить существует ли в этой таблице фамилия ИГУМНОВ. Предположим, что мы еще никаких индексов на эту таблицу не сделали, так же фамилия ИГУМНОВ стоит посередине таблице. Когда мы пошлем вот такой запрос: select surname from ourtable where surname='ИГУМНОВ' база данных переберет пол миллиона записей пока не дойдет до фамилии ИГУМНОВ и не выдаст результат. Получается слишком медленно. Но как только мы сделаем индекс на поле нашей таблицы, как сразу все наши запросы будут обрабатываться за миллисекунды, чего мы и добиваемся. Естественно, одной таблицы будет мало для решения нашей проблемы.

Начнем с таблицы document. В этой таблице хранятся имена файлов или URL'ы страниц и каждой такой записи сопоставлен уникальный ключ id. В таблице dictionary хранятся все слова, которые могут встретиться в наших документах, и каждому слову сопоставлен уникальный id. Естественно, создаются индексы на поле word в таблице dictionary и на поле id в таблице document. В нашем примере существует отношение многие ко многим. Это необходимо, так как в таблице match мы храним соответствие слова и документа. Другими словами, в таблице match хранится информация о том, какие слова есть в каждом документе. На таблицу match создают индекс, на поле dict\_id.

**Индексный механизм**

Прежде чем ваши документы будут доступны для поиска, их необходимо проиндексировать. Объем индексной информации, полученной из текста, может быть в два раза больше чем сам тексте. А может еще больше, в случае если вы будете не оптимально использовать память. Алгоритм выглядит следующим образом:

1. получаем документ для индексирования;

2. регистрируем его в таблице document, запоминаем полученный его уникальный id и будем его называть doc\_id;

3. разбиваем документ на отдельные слова;

4. узнаем уникальные id этих слов из таблицы dictionary и будем их называть dict\_id;

5. потом заносим записи с нашим одним doc\_id и разными dict\_id (для каждого слова в документе) в таблицу match.

**Поисковый механизм**

После того как мы проиндексировали наши документы, нужно понять какие запросы посылать в базу, что бы искать эти документы по ключевым словам. Предположим, есть поисковая фраза "река объ". Пользователю необходимо получить все документы содержащие эти два слова. Сначала нужно обратиться к таблице dictionary и узнать уникальные id этих слов, далее будем их называть $dict\_id1 и $dict\_id2. Потом необходимо послать такой запрос в таблицу match, который выдаст только те номера документов, которые содержат эти два слова. Вот пример этого запроса: SELECT doc\_id FROM match where dict\_id =$dict\_id1 group by doc\_id INTERSECT SELECT doc\_id FROM match where dict\_id=$dict\_id2 group by doc\_id. В случае если пользователь введет три слова, то вам придется добавить еще раз INTERSECT и третью часть SQL запроса. По полученным в результате запроса doc\_id можно извлечь информацию об имени файла документа из таблицы document.

**Комплексное функционирование**

Существует три потока управления. Первый обслуживает запросы пользователя, второй выполняет поисковые запросы, а третий занимается индексированием новых документов поступающих в систему. Первый поток - это скрипт на Perl, Servlet, ASP или PHP, который из ключевых слов пользователя формирует поисковые SQL запросы. Второй поток - это СУ базой данных, которая поддерживает целостность данных, индексный механизм и обслуживает SQL запросы. Третий поток - это тоже скрипт, который работает с новыми документами, индексирует их и посылает запросы в базу данных на внесения новой индексной информации.

**Принципы работы поисковой машины Рамблер**

Интернет постоянно растет, так же как растет и число пользователей, которые обращаются с запросами к поисковым системам. Увеличение объема информации и количества запросов, в свою очередь, приводит к повышению требований к скорости работы поисковых машин, качеству поиска и наглядности представления результатов. Так, для того чтобы пользователь остался доволен результатом, на сегодняшний день поисковой системе нужно собрать, обработать, обновить, найти и отсортировать в два раза больше документов, чем год назад. А основная задача поиска как раз и состоит в том, чтобы пользователь был доволен его результатами.

Когда пользователь обращается с запросом к поисковой машине, он хочет найти то, что ему нужно, максимально быстро и просто. Получая результат, он оценивает работу системы, руководствуясь несколькими основными параметрами. Нашел ли он то, что искал? Если не нашел, то сколько раз ему пришлось переформулировать запрос, чтобы найти искомое? Насколько актуальную информацию он смог найти? Насколько быстро обрабатывала запрос поисковая машина? Насколько удобно были представлены результаты поиска? Был ли искомый результат первым или сотым? Как много ненужного мусора было найдено наравне с полезной информацией? Сможет ли он, вернувшись завтра и дав тот же запрос, получить те же результаты?

Для того, чтобы ответы на эти вопросы оставались хоть немного удовлетворительными, разработчики поисковых машин постоянно совершенствуют алгоритмы и принципы поиска, добавляют новые функции, ускоряют работу системы. В этой статье мы обратимся к механизму работы поисковой машины Рамблер, и на примере ее устройства продемонстрируем, как достигается повышение качества и скорости поиска в условиях постоянного роста объема информации в сети Интернет.

Полнота - это одна из основных характеристик поисковой системы, которая представляет собой отношение количества найденных по запросу документов к общему числу документов в Интернете, удовлетворяющих данному запросу.

Полнота поиска в большой мере зависит от работы системы сбора и обработки информации. В связи с постоянным ростом количества документов в сети, эта система в первую очередь должна быть масштабируемой. В Рамблере масштабируемость достигается за счет параллельного исполнения задачи произвольным количеством машин.

Сбором информации занимается робот-паук, который обходит страницы с заданными URL и скачивает их в базу данных, а затем архивирует и перекладывает в хранилище суточными порциями. Робот размещается на нескольких машинах, и каждая из них выполняет свое задание. Так, робот на одной машине может качать новые страницы, которые еще не были известны поисковой системе, а на другой - страницы, которые ранее уже были скачаны не менее месяца, но и не более года назад. Хранилище у всех машин едино. При необходимости работу можно распределить другим способом, например, разбив список URL на 10 частей и раздав их 10 машинам. Параллельная работа программы позволяет легко выдерживать дополнительную нагрузку: при увеличении количества страниц, которые нужно обойти роботу, достаточно просто распределить задачу на большее число машин.

После того, как все части информации обработаны, начинается объединение (слияние) результатов. Благодаря тому, что частичные индексные базы и основная база, к которой обращается поисковая машина, имеют одинаковый формат, процедура слияния является простой и быстрой операцией, не требующей никаких дополнительных модификаций частичных индексов. Основная база участвует в анализе как одна из частей нового индекса.

Сборка единой базы из частичных индексных баз представляет собой простой и быстрый процесс. Сопоставление страниц не требует никакой интеллектуальной обработки и происходит со скоростью чтения данных с диска. Если информации, которая генерируется на машинах-индексаторах, получается слишком много, то процедура "сливания" частей проходит в несколько этапов. Вначале частичные индексы объединяются в несколько промежуточных баз, а затем промежуточные базы и основная база предыдущей редакции пересекаются. Таких этапов может быть сколько угодно. Промежуточные базы могут сливаться в другие промежуточные базы, а уже потом объединяться окончательно. Поэтапная работа незначительно замедляет формирование единого индекса и не отражается на качестве результатов.

Точность - еще одна основная характеристика поисковой машины, которая определяется как степень соответствия найденных документов запросу пользователя. Например, если по запросу "Красная площадь" находится 150 документов, в 70 из них содержится словосочетание "Красная площадь", а в остальных просто присутствуют эти слова ("красная баба кричала на всю площадь"), то точность поиска считается равной 70/150 (~0,5). Чем точнее поиск, тем быстрее пользователь находит нужные ему документы, тем меньше "мусора" среди них встречается, тем реже найденные документы не соответствуют запросу.

Способ повышения точности поиска - это выделение устойчивых обозначений и поиск их как отдельных лексических единиц. На сегодняшний день в Рамблере реализована система распознавания таких конструкций, например C++, б/у, п/п-к. Если по запросу С++ поднимать все тексты, в которых присутствуют латинская буква С, а также знак +, то получится огромное количество документов, далеко не все из которых соответствуют запросу; кроме того, это большая работа, значительно увеличивающая время поиска.

Помимо автоматических способов увеличения точности поиска, существуют различные средства, с помощью которых пользователь сам может уточнить поиск по отдельным запросам. В первую очередь к ним относится специальный язык поискового запроса, используя который можно ограничивать количество найденных документов. Например, запрос или его часть, взятые в кавычки, обрабатываются буквально, с учетом всех стоп-слов, форм, порядка, знаков препинания. Это повышает точность поиска, но уменьшает его полноту: если часть, заключенная в кавычки, неточна, нужный документ найден не будет.

Актуальность - не менее важная характеристика поиска, которая определяется временем, проходящим с момента публикации документов в сети Интернет, до занесения их в индексную базу. Например, на следующий день после теракта в Тушино огромное количество пользователей обратились к поисковой машине Рамблер с соответствующими запросами. Объективно с момента публикации новостной информации на эту тему прошло меньше суток. Однако основные документы уже были заиндексированы и доступны для поиска, благодаря существованию "быстрой базы", которая обновляется два раза в день, а при необходимости может обновляться быстрее.

Скорость поиска тесно связана с его устойчивостью к нагрузкам. На сегодняшний день в рабочие часы к поисковой машине Рамблер приходит около 60 запросов в секунду. Такая загруженность требует сокращения времени обработки отдельного запроса. Здесь интересы пользователя и поисковой системы совпадают: посетитель хочет получить результаты как можно быстрее, а поисковая машина должна отрабатывать запрос максимально оперативно, чтобы не тормозить вычисление следующих.

Запрос поступает в поисковую систему через маршрутизатор Cisco 6000 series. Cisco передает его наименее загруженной машине первого уровня - frontend (1.1 - 1.3, на рис. машине 1.3). Frontend, в свою очередь, отправляет запрос дальше, на один из восьми proxy-серверов, также выбирая наиболее свободный сервер (2.1 - 2.8, на рис. машине 2.2). Одновременно frontend отправляет запрос на машины, осуществляющие поиск по товарам (3.1 - 3.2, машине 3.1) и по базе Тор 100 (4.1 - 4.2, машине 4.1). На proxy проводится поиск по ссылочному индексу, и его результаты вместе с поисковым запросом передаются на машины, которые содержат основную индексную базу, - backends (5.1.х - 5.7.х, машинам 5.1.2, 5.2.11, 5.3.1 и т.д.). Та же информация отправляется на машины с "быстрой базой" (6.1 - 6.2).

На текущий момент в поиск включено 77 backend'ов. Они сгруппированы по 11 машин, и каждая группа содержит копию одной из частей поискового индекса. Таким образом, информация о сайтах, условно входящих в красный сектор Интернета, находится на backend'ах первой группы (5.1.1 - 5.1.11), оранжевый сектор - на backend'ах второй группы (5.2.1 - 5.2.11) и т.д. Proxy-сервер выбирает наименее загруженный backend в каждой группе машин и отправляет на него поисковый запрос с результатами ссылочного поиска. На backend'ах осуществляется поиск по частям индексной базы и ранжирование с учетом результатов поиска по ссылочному индексу. При ранжировании для всех найденных документов высчитываются веса по конкретному запросу.

После того, как запрос обработан на backend'ах, информация о результатах и ранжировании отдается обратно на proxy-сервер. Туда же поступают отсортированные результаты с машин "быстрой базы". Proxy интегрирует данные, полученные с восьми машин: клеит дубли, объединяет зеркала сайтов, переранжирует документы в общий список по весам, рассчитанным на backend'ах. Так, первым в списке найденного может быть документ с машины 5.3.1, вторым и третьим - с 6.1, четвертым - с 5.5.2 и т.д. На proxy-сервере также реализуется построение цитат к документам и подсветка слов запроса в тексте. Полученные результаты отдаются на frontend.

Помимо информации с proxy-сервера, frontend получает результаты из поиска по товарам и из базы Тор 100, отсортированные, с цитатами и подсветкой слов запроса. Frontend осуществляет окончательное объединение результатов, генерирует html со списком найденного, вставляет баннеры и перевязки (ссылки на различные разделы Рамблера) и отдает html Cisco, который маршрутизирует информацию пользователю.

Каждый из этапов обработки запроса многократно продублирован и защищен системой балансировки нагрузки. Благодаря дублированию информации поисковая система Рамблер является устойчивой к сбоям на отдельных участках, авариям, отказам оборудования. Если одна их машин перестала функционировать, нагрузка перераспределяется на другие машины, и выпадения документов из поиска не происходит. Масштабируемость достигается простым добавлением в систему машин соответствующего уровня. До недавнего времени в Рамблере работало 45 backend'а. В связи с тем, что осенью нагрузка на поисковые системы обычно возрастает, число backend'ов было увеличено до 77, что позволило значительно ускорить вычисление запросов.

***Консультация №2***

**3.Технология управления проектами на примере Project expert и Microsoft Project**

**Использование MS Project для управления проектами по разработке ПО**

**Введение**

Вся методология — это просто набор простых методов и рекомендаций по использованию MS Project для решения прикладных задач руководителя проекта. Сразу оговорюсь, что методология не претендует на универсальность, и применима только при некоторых ограничениях, которые я буду упоминать по ходу повествования.

Для начала, давайте вспомним, что обычно требуется от руководителя проекта. Для опытных руководителей это очевидно, а начинающим (или только собирающимся стать руководителями) будет полезно лишний раз вспомнить. Итак, проект по разработке программного обеспечения — это создание некоторое уникального продукта. На разных этапах жизненного цикла проекта от РП требуется решать различные задачи.

**Перед началом проекта**

Перед началом проекта от руководителя проекта обычно требуется ответить на два вопроса:

1. сколько проект займет времени
2. сколько проект будет стоить

При этом важно понимать, что никого не интересует ответ вида «не раньше чем через полгода». Требуется как раз оценка сверху.

**В процессе выполнения проекта**

В условиях упомянутых ограничений, основной задачей руководителя проекта является обеспечить выполнение проекта в заявленный срок, а это непосредственно  
влияет на его стоимость. Непредвиденные обстоятельства, которые обязательно сопутствуют любому проекту, могут привести к срыву сроков. Строго говоря, сроки проекта могут неожиданно и сократиться, но, честно говоря, я такого никогда не видел. От руководителя требуется своевременно реагировать на такие события, чтобы уменьшить негативные последствия. Единственный известный мне способ решения этой задачи — это аккуратное планирование, регулярное отслеживание надвигающихся проблем и корректирование планов.

**При завершении проекта**

При завершении проекта руководитель обычно оглядывается назад и подводит итоги проекта. Чаще всего требуется оценить насколько проект выбился из плановых графиков и почему это произошло.

**Что умеет MS Project**

Несмотря на внешнюю сложность, MS Project очень прост в идейном плане. Он оперирует тремя сущностями — задачи, ресурсы, календарь и связи между ними. По сути — это база данных, пользовательский интерфейс для создания и редактирования сущностей и минимальная, довольно простая автоматизация (то, что Project делает сам, в ответ на введенные данные).

Разберем вкратце свойства сущностей.

**Задача** имеет длительность, объем, назначенный ресурс и еще чертову уйму различных свойств. Если встроенных свойств не хватает, можно добавить свои — этим мы потом воспользуемся. Задачи могут быть связаны между собой различными отношениями (предшественники, последователи и т.п.).

**Ресурс** имеет много описательных свойств, но самое главное — для него можно  
задать доступность во времени, для этого используется календарь. Ресурс может быть  
назначен на задачу.

На основе этих данных Project умеет делать различные представления с использованием фильтров, группировок, сортировок и т.п. Кроме этого он умеет по некоторому алгоритму вычислять сроки начала и окончания задач с учетом доступности назначенных ресурсов и связей между задачами. Вот, собственно, и почти все что он умеет.

Давайте посмотрим, какую пользу можно из этого извлечь

**Как это использовать**

**Подготовка плана**

Итак, перед нами лежит техническое задание, и требуется дать ответ на три вопроса:

1. Сколько времени займет этот проект?
2. Сколько (и каких) специалистов для этого потребуется?
3. Какие примерно трудозатраты ожидаются по этому проекту?

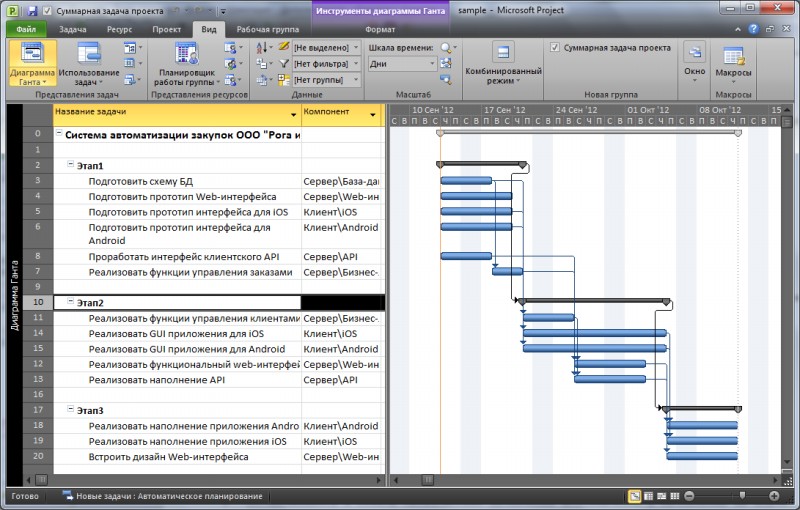
Для этого мы готовим прикидочный план выполнения проекта в MS Project. Т.е. просто последовательно выписываем задачи, которые необходимо выполнить. Методика превращения техзадания в набор задач — это отдельная история, я не буду на ней сейчас останавливаться.  
Подготовка плана выполняется в несколько этапов:

1. Готовим список задач
2. Выставляем зависимости между задачами  
   (результат какой задачи необходим для перехода к следующей?).
3. Назначаем исполнителей задач
4. Выравниваем загрузку ресурсов
5. Балансируем то, что получилось

**Общие рекомендации**

При подготовке плана придерживаемся следующих рекомендаций:

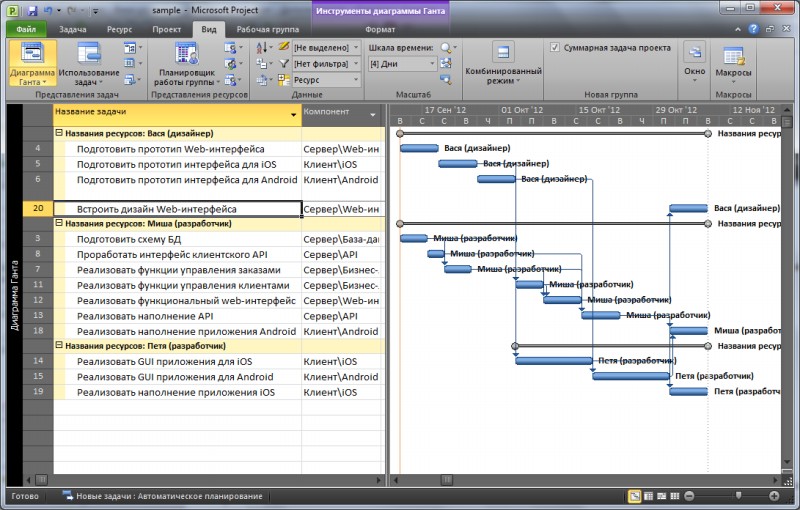
1. Не используем суммарные задачи для декомпозиции.  
   Все задачи помещаем в один линейный список. Сначала это может показаться неудобным,  
   но зато избавляет от многих проблем в дальнейшем. Для управления структурой задач  
   используем настраиваемые поля (см.ниже).
2. Очень часто для управления зависимостями задач используют Drag&Drop. Когда задач много это быстро становится неудобно. Я рекомендую в этом случае не использовать перетаскивание, а явное указывать номера задач-предшественников. для этого можно добавить в таблицу столбец «предшественники» и вписывать номера задач вручную.
3. Срок каждой задачи не должен превышать двух недель.  
   Если срок задачи превышает неделю — это уже повод задуматься о её декомпозиции. Я придерживался очень простой методики оценки: примитивная задача — 2 дня, средней  
   сложности — 1 неделя, сложная задача — 2 недели. При этом сложных задач не должно быть много. Такой подход дает возможность подготовить оценочный план довольно быстро.
4. С одной стороны, полученная оценка, конечно, не будет точной, но, с другой стороны — а какая из них точная? По опытку практического применения могу сказать, что на  
   больших проектах погрешности оценок отдельных задач обычно нивелируются, а на малых часто можно (и нужно!) использовать и более точные оценки.
5. Всеми силами избегаем задач, у которых несколько исполнителей. Для каждой задачи должен быть назначен только один исполнитель. Двух исполнителей имеет смысл назначать  
   только если они действительно работают вдвоем (например, вы практикуете парное программирование). В прочих случаях лучше декомпозировать задачу.
6. При назначении исполнителей руководствуемся их профессией и квалификацией, пока не беспокоясь о равномерности загрузки.
7. Используем суммарные задачи для разделения задач на этапы. Ставим зависимости между этапами, чтобы они шли последовательно. Разделение на этапы пока достаточно приблизительное.

  
Список задач, разделенный на этапы

**Балансировка проекта**

Самым главным в методике является именно балансировка. Цель этого процесса — подготовить план, в котором работы достаточно равномерно разделены между исполнителями на всем протяжении.

После первичной подготовки плана обычно получается полное безобразие, а не проект. Поэтому начинаем приводить его в порядок. Приведение в порядок заключается в ручной балансировке назначений исполнителей и разделений на этапы. Для этого используем **группировку задач по исполнителям**, чтобы увидеть как разложились задачи. Для удобства просмотра рекомендую сортировать задачи по дате начала.

  
Группировка задач по исполнителям

**Примечание.** Теоретически, для оценки загрузки полагается использовать график загрузки пользователей. Эти графики хороши (наверное) для начальства, когда они  
оценивают готовый проект. Но они непригодны на этапе создания плана, так как показывают что все плохо, но совершенно не дают информации почему это так и что можно сделать.

Дальше начинается магия балансировки. Требуется минимизировать сроки выполнения каждого этапа путем обеспечения более-менее равномерной нагрузки на всех участников проекта. Для этого мы выполняем следующие действия:

1. Сменить исполнителя задачи.
2. Это имеет смысл сделать, если мы видим, что у одного исполнителя большой хвост задач,  
   а у другого есть явные «дыры», причем он может взять на себя некоторые работы у  
   первого.
3. Перенести задачу в другой этап.

Задача, которая приводит у удлинению срока этапа, но при этом не является необходимой для получения результата этапа может быть перенесена на этап позже. И наоборот,  
если в этапе присутствуют «дыры» в загрузке исполнителей, а изменить исполнителей  
не получается, то можно попробовать взять задачи из следующего этапа.

Делать все это, к сожалению, приходится вручную, выполняя выравнивание загрузки ресурсов после каждого изменения. Несмотря на кажущуюся сложность, этот процесс обычно занимает конечное время. Проект на год из 8 участников, разбитый на 4 этапа я приводил в порядок менее чем за час.

Теперь еще раз внимательно смотрим на проект, убеждаемся, что связи между задачами расставлены правильно, что ничего не забыто, а назначения исполнителей соответствуют их специальностям и квалификации.

**Учет рисков**

Теперь — последний штрих: учет рисков. Честно признаюсь, я не занимался серьезным управлением рисками, но учитываю возможность возникновения определенных форсмажоров (таких как болезни исполнителей, забытые работы и т.п.). Для этого я добавляю в каждый этап фиктивную задачу с минимальным приоритетом, под названием «прочие работы» для каждого ресурса. После выравнивания ресурсов эти задачи оказываются в конце этапа. Длительность этих задач зависит от вероятности возникновения и степени вляния рисков, она зависит от способа определения оценок длительностей задач, здоровья членов команды и степени паранойи руководителя проекта. Обычно я выставлял длительность «прочих работ» примерно от трети до четверти длины этапа.

В результате всех перечисленных манипуляций у нас получается план выполнения проекта, с которым можно работать.

С этим планом мы можем:

1. Назвать сроки выполнения проекта и его этапов. Аргументированно и с высокой степенью  
   достоверности.
2. Оценить примерные трудозатраты по проекту

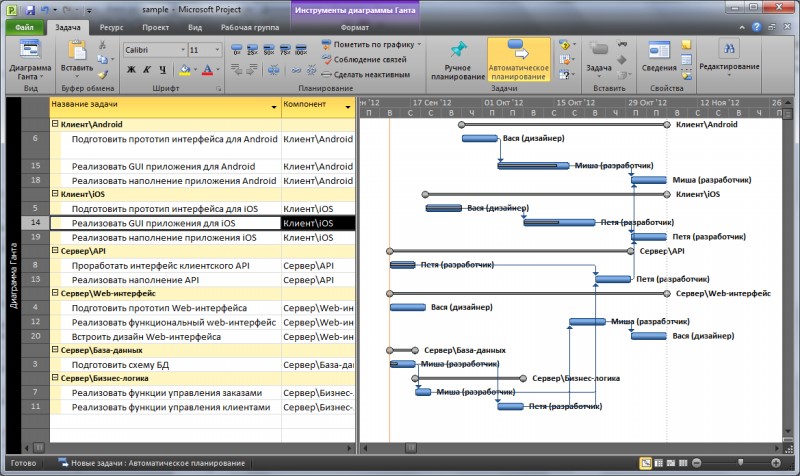
**Примечание.** Часто случается так, что срок выполнения получается довольно большой, и возникает резонный вопрос, можно ли его уменьшить за счет привлечения дополнительных исполнителей. Для того чтобы ответить на этот вопрос, я балансировал новый план, используя тот же набор задач, но изменяя состав исполнителей. Ответ не получался мгновенно, но это не занимало много времени.

Работа с планом

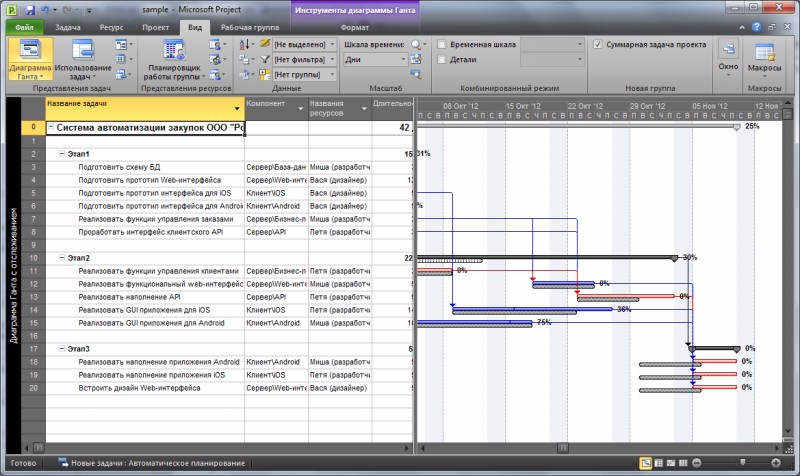
Когда проект запускается в работу, исходный план, который использовался для оценки, можно использовать и для отслеживания выполнения проекта. От руководителя проекта требуется регулярно выполнять следующие действия:

1. Выдавать задания исполнителями
2. Отмечать выполненные задания в плане
3. Корректировать план в случае значительных отклонений

Выдача заданий исполнителями может выполняться по разному. Можно разбить выполнение на короткие итерации, формировать пул задач на итерацию и по окончании итерации отмечать результаты. Можно сразу озвучить лнителям набор задач на этап, выдать каждому по экземпляру диаграммы Ганта и периодически опрашивать о прогрессе. Можно использовать интеграцию MS Project и TFS и загрузить проект непосредственно в TFS. Суть не в средствах. Главное — это **регулярное обновление плана**. Я делаю это примерно раз-два в неделю. Это дает возможность достаточно быстро увидеть проблемные участки.  
Для определения проблемного участка удобно использовать различные группировки — по исполнителями, по компонентам и др. Часто может оказаться, что проект в целом идет даже с опережением, но в определенном разрезе наблюдается отставание, например один из разработчиков неожиданно уткнулся в серьезную системную проблему, которая привела к отклонениями. Использование только средней метрики не покажет этой проблемы — она всплывет только в конце этапа, когда что либо делать будет уже поздно.

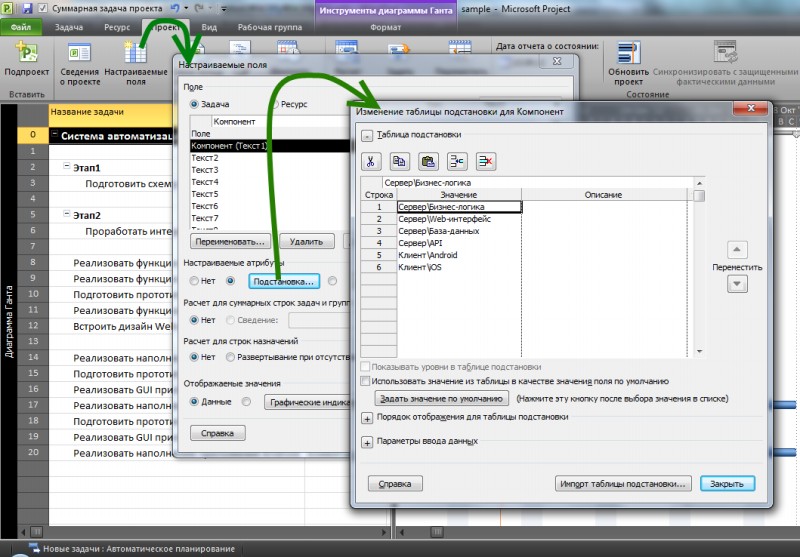
  
Отслеживание выполнения с группировкой по компонентам

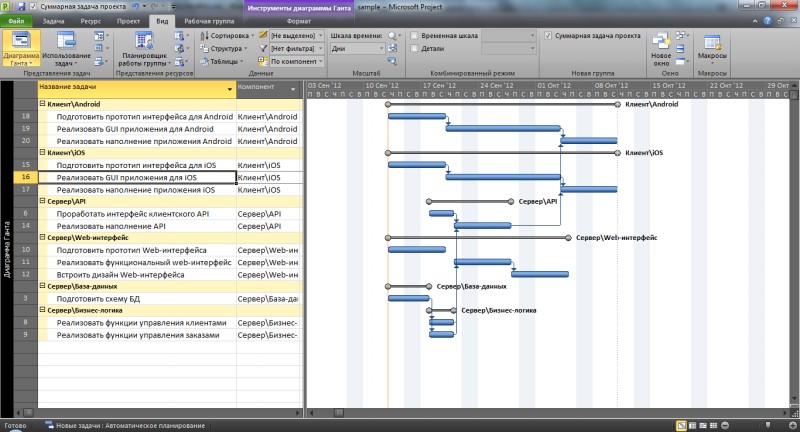
Есть другая стратегия — внесение изменений в сроки задач, «выталкивая» невыполненные задачи вперед. При таком подходе для отслеживания отклонений от плана можно использовать другую полезную функцию MS Project — базовый план. Базовый план — это просто сохраненный снимок состояния задач. Его можно сделать в начале проекта. Для сравнения текущего плана с базовым, открываем «диаграмму Ганта с отслеживанием». Для динамичного плана, когда порядок выполнения задач часто меняется, это может оказаться неудобным, поэтому я вставляю в проект контрольные точки, отражающие некоторые важные результаты проекта, и отслеживать отклонения от базового плана только для них.

  
Диаграмма Ганта с отслеживанием

**Управление структурой задач с помощью пользовательских полей**

Я категорически рекомендую не использовать суммарные задачи в MS Project для функциональной декомпозиции или категоризации задач. Дело в том, что иерархия задач в MS Project сильно завязана на их последовательность. А часто хочется посмотреть на задачи в разной последовательности, при этом вся структура «рассыпается». Для управления структурой задач я рекомендую использовать **Пользовательские поля**. MS Project имеет предопределенный набор полей с неопределенным заранее поведением, которые мы можем использовать так, как нам удобно. Например, для разбивки задач по компонентам нужно на основе текстового поля **Текст1** создать поле **Компонент** и задать для него список значений, соответствующий компонентам системы.

  
Создание пользовательского поля

После этого мы получаем возможность указать для каждой задачи компонент, к которому она относится, и, используя группировку задач по компонентам, отслеживать как идут дела.  
  
  
Группировка задач по компонентам

Пользовательские поля позволяют разделять задачи по нескольким категориям, например, разделять задачи по типу работ: Разработка, Тестирование, Документирование.  
*В MS Project также можно задать правила рисования диаграмм на основе свойств задач. При желании, можно сделать так, что задачи по разным компонентам будут иметь разные цвета, причем цвет будет определяться только свойством задачи, его не нужно задавать вручную для каждой задачи. Такие настройки не требуют написания сриптов, а делаются штатными средствами настройки диаграмм.*

Использование пользовательских полей, а также встроенные в MS Project функции фильтрации, сортировки и группировки задач позволяют получить самые разные представления, которые позволяют получить ответы на многие вопросы, которые возникают у руководителя проекта.

**Завершение проекта**

В конце проекта мы получаем план, в котором все задачи выполнены. Обычно я стараюсь сохранять также и исходный план, хотя бы в качестве базового. Честно говоря, на этом этапе от MS Project мало проку, так как интересуют не плановые значения, а фактические. Какие-то решения этой проблемы предлагает MS Project Server, там есть возможность учитывать фактические трудозатраты, но это уже за пределами данной статьи.

**Заключение**

Описанная методика не претендует не универсальность, но она достаточно проста и логична, при этом позволяет решать практические задачи руководителя проекта.

Использование этого подхода позволило успешно и в срок завершить не один проект. Правда, случались и сбои. Это происходило, как правило, тогда, когда плохо была проведена подготовительная часть проекта, а именно — постановка задачи. Т.е. в результате проекта получалось не совсем то, что требовалось, а понимание этого приходило слишком поздно.

**4. Технология комплексной автоматизации деятельности предприятия на примере 1С:Предприятие 8.3**

**Задание1. ПРИОБРЕТЕНИЕ МАТЕРИАЛОВ ЗА ПЛАТУ в программе «1С-Бухгалтерия».**

**Задача 1**. Предприятие «МАРИНА» по счету №1245/28 от ( студент проставляет текущую дату) , поступившему от фирмы «ОРИОН» ,произвело предоплату «Ориону» в счет поставки материалов согласно прилагаемой таблице. на сумму 137500 руб., включая НДС 18%. Выписано платежное поручение №2 от (текущая дата) и отправлено в КБ «Мегаполис». Пиломатериалы поступили в «МАРИНУ» в сопровождении товарно– транспортной накладной и счета – фактуры № 1854/87. Пиломатериалы прибыли на склад материалов и оприходованы, т.е. кладовщиком составлен приходный ордер.

Факт предоплаты, поступления и передачи материалов необходимо отразить в бухучете.

Отразить этот факт в бухучете.

ОПЛАЧЕННЫЕ МАТЕРИАЛЫ

Единица измерения

Получено в количестве

Цена

Стоимость

1

Плита ДСП 2х1.2

Шт.

100

250

25 000

2

Брус осиновый

Куб.м.

40

1500

60 000

3

Шпон дубовый 1200мм.

М.

25

950

23 750

4

Шпон ореховый 1000мм

М.

30

800

24 000

5

Лак ЛМП-766

литр

10

50

500

6

Морилка спиртовая темная

Литр

10

50

500

7

Плита ДСП 1х0.6

Шт.

25

150

3 750

ИТОГО

137 500

Для оплаты выбираем Меню БАНК-ПЛАТЕЖНОЕ ПОРУЧЕНИЕ ИСХОДЯЩЕЕ. Оформляем на сумму 137500 руб. Далее оформляем приобретение меню ПОКУПКА-ПОСТУПЛЕНИЕ ТОВАРОВ И УСЛУГ, Товаром являются Материалы. После проведения оформить СЧЁТ-ФАКТУРУ.

**Задача 2.** Основному производству по требованию – накладной №1 от (текущая дата) переданы со склада материалов для производства офисной мебели следующие материалы:

Счет затрат

Единица измерения

Отпущено количество

Цена

1

Плита ДСП 2х 1.2

20

Шт.

50

250

2

Брус осиновый

20

Куб. м.

25

1500

3

Шпон дубовый 1200 мм

20

М.

15

950

4

Шпон орех 1000мм

20

М.

20

800

5

Лак ЛМП –766

20

Л.

5

50

6

Морилка спиртовая темная

20

Л.

5

50

7

Плита ДСП 1.0х0.6

20

Шт.

10

150

Отразить в бухучете факт передачи материалов в основное производство. Зайти в Меню ПРОИЗВОДСТВО-ТРЕБОВАНИЕ\_НАКЛАДНАЯ.

**Задание 2. ПЕРЕДАЧА ГОТОВОЙ ПРОДУКЦИИ НА СКЛАД в программе «1С-Бухгалтерия».**

**Задача 3**. На склад готовой продукции предприятия «МАРИНА» по документу ПРИЕМО – СДАТОЧНАЯ НАКЛАДНАЯ №1 от (текущая дата) переданы из производства готовые изделия. В приемо – сдаточной накладной отражена следующая информация:

Единица измерения

Количество

Плановая себестоимость

1

Стол обеденный

Шт.

5

700

2

Стол – книжка

“

5

650

3

Стол кухонный светлый

“

10

850

4

Стол кухонный темный

“

10

753

5

Стол « директорский»

“

7

900

6

Стол «клерк»

“

10

500

Приемо – сдаточная накладная передана в бухгалтерию. Отразить факт выпуска готовой продукции в компьютерной бухгалтерии.

Решение. Для оформления операций, связанных с передачей готовой продукции на склад в типовой конфигурации предназначен документ Отчёт о реализации за смену. Меню ПРОИЗВОДСТВО- ОТЧЁТ О РЕАЛИЗАЦИИ ЗА СМЕНУ.

**Задание 3. ОТГРУЗКА ГОТОВОЙ ПРОДУКЦИИ в программе «1С-Бухгалтерия».**

**Задача 4.** Предприятие «МАРИНА» выписало счет №1 от текущей даты заводу «КОЛИБРИ» на оплату следующей продукции:

Стол «Директорский – 5 шт. по цене 1500 руб.

Стол «Клерк» – 10 шт. по цене 1000 руб.

По отпускной цене с 18% НДС.

Одновременно выписана накладная на отпуск продукции со склада представителю завода «Колибри» (на основании имеющейся у него доверенности). Необходимо реализовать эти действия по формированию и печати названных документов.

Для выписки счёта меню «ПРОДАЖА» - «СЧЕТ НА ОПЛАТУ ПОКУПАТЕЛЮ».Заполним выставленный счет. Документ «Счет на оплату покупателю» не совершает никаких проводок в бухгалтерском учете.

На основании выписанного счета оформим оплату ПЛАТЕЖНЫМ ПОРУЧЕНИЕМ ВХОДЯЩИМ. На основании выписанного счета введем новый документ РЕАЛИЗАЦИЯ ТОВАРОВ И УСЛУГ. После проведения данного документа оформим СЧЁТ-ФАКТУРУ выданную.

**Задание 4 АВАНСОВЫЕ ОТЧЁТЫ в программе «1С-Бухгалтерия».**

**Задача 5.** Сотрудник Попов А.А. получил под отчёт 7000 руб. 16 числа текущего месяца. 17 числа текущего месяца на Общий склад оприходовано 60 пачек офисной бумаги, в бухгалтерию представлен авансовый отчёт и чек №876 от ТД «Премьер офис» на сумму 6600 руб.

Оформим Расходный Кассовый ордер (меню КАССА-РАСХОДНЫЙ КАССОВЫЙ ОРДЕР) на выдачу денег подотчёт от 16.\*\*.10. От 17.\*\*.10г. оформляем Авансовый отчёт (КАССА- АВАНСОВЫЙ ОТЧЁТ) назначение- на приобретение канц. Товаров, не учитываем НДС, на вкладке «Товары» указываем офисную бумагу.

**Задача 6**. Сотрудник Яньков В.Ю. получил под отчёт 4500 руб. на командировочные расходы 16 числа текущего месяца. 22 числа текущего месяца в бухгалтерию представлен авансовый отчёт, в котором отражены расходы:

билет Чебоксары – Казань 500 руб.,

билет Казань - Чебоксары 500 руб.,

счёт гостиницы на 2360 руб.( в т.ч. НДС(18%) - 360 руб., предъявлена счёт-фактура №123 от 17.\*\*.\*\*)

суточные за 6 суток -600 руб.

Оформим Расходный Кассовый ордер (меню КАССА-РАСХОДНЫЙ КАССОВЫЙ ОРДЕР) на выдачу денег подотчёт от 16.\*\*.10. От 22.\*\*.10г.оформляем Авансовый отчёт (КАССА- АВАНСОВЫЙ ОТЧЁТ) назначение - на командировочные расходы, не учитываем НДС(кроме счёта гостиницы), на вкладке «ПРОЧЕЕ» указываем все расходы..

**Задание 5 ЗАРПЛАТА в программе «1С-Бухгалтерия».**

**Задача 7.** Начислить зарплату всем сотрудникам «МАРИНЫ» за текущий месяц.

1.В Меню ЗАРПЛАТА Открываем окно НАЧИСЛЕНИЕ ЗАРПЛАТЫ. Убедимся, что в графе ДАТА проставлен последний день текущего месяца или установим его.

2.Нажмем кнопку ЗАПОЛНИТЬ. Все сотрудники, занесенные в СПИСОК СОТРУДНИКОВ, с их окладами будут занесены в таблицу.

**Задача 8.** Составить платежную ведомость выплаты начисленной зарплаты за текущий месяц. Выплатить зарплату через кассу, предварительно получив в банке деньги на выплату зарплаты. Рассчитать взносы в фонды с ФОТ. Отразить зарплату в регламентированном учёте. Составить расчётные листки на каждого работника. Составить расчётную ведомость.

Меню ЗАРПЛАТА-ЗАРПЛАТА К ВЫПЛАТЕ. Выбрать команду «Заполнить» из меню документа и ее вкладку «По задолженности на конец месяца», а затем команду «Рассчитать». Данный документ имеет ряд печатных форм, одна из которых Т-49 «Расчетно-платежная ведомость», другая Т-53 «Платежная ведомость». Сформируем «Расчетно-платежную ведомость».

Для осуществления проводок в бухгалтерском плане счетов служит документ «Отражение заработной платы в регламентированном учете», по которому происходят изменения в синтетическом учете предприятия. Данный документ оформляется после создания документов «Начисление заработной платы» и «Расчет страховых взносов» за месяц, данные для проводок берутся из этих, предшествующих ему, документов. Найти данный документ можно следующим образом: главное меню вкладка «ЗАРПЛАТА» - «ОТРАЖЕНИЕ ЗАРАБОТНОЙ ПЛАТЫ В РЕГЛАМЕНТИРОВАННОМ УЧЕТЕ».

**Задание 6. ОПЕРАЦИИ, ПРОВОДИМЫЕ В ПОСЛЕДНИЙ ДЕНЬ МЕСЯЦА (РЕГЛАМЕНТНЫЕ) в программе «1С-Бухгалтерия». УЧЁТ НДС**

1. Сформировать Книгу продаж (Меню ПРОДАЖА-ВЕДЕНИЕ КНИГИ ПРОДАЖ-ФОРМИРОВАНИЕ ЗАПИСЕЙ КНИГИ ПРОДАЖ, ПРОДАЖА-ВЕДЕНИЕ КНИГИ ПРОДАЖ - КНИГА ПРОДАЖ).
2. Сформировать Книгу покупок (Меню ПРОДАЖА-ВЕДЕНИЕ КНИГИ ПОКУПОК-ФОРМИРОВАНИЕ ЗАПИСЕЙ КНИГИ ПОКУПОК, ПРОДАЖА-ВЕДЕНИЕ КНИГИ ПОКУПОК- КНИГА ПОКУПОК).
3. Сформировать декларацию по НДС за месяц: ОТЧЁТЫ-РЕГЛАМЕНТИРОВАННЫЕ: - НАЛОГОВАЯ ОТЧЁТНОСТЬ-НДС. Расчёт общей суммы налога- выбрать месяц, нажать кнопку ЗАПОЛНИТЬ.
4. Произвести последней датой месяца «Закрытие месяца» (меню ОПЕРАЦИИ - РЕГЛАМЕНТНЫЕ ОПЕРАЦИИ - ЗАКРЫТИЕ МЕСЯЦА).
5. Сформировать бухгалтерскую отчётность: БАЛАНС И ОТЧЁТ О ПРИБЫЛЯХ И УБЫТКАХ.
6. ОТЧЁТЫ - РЕГЛАМЕНТИРОВАННЫЕ: - БУХГАЛТЕРСКАЯ ОТЧЁТНОСТЬ - БУХГАЛТЕРСКИЙ БАЛАНС. Выбрать месяц, нажать кнопку ЗАПОЛНИТЬ.
7. ОТЧЁТЫ - РЕГЛАМЕНТИРОВАННЫЕ: - БУХГАЛТЕРСКАЯ ОТЧЁТНОСТЬ – Отчёт о прибылях и убытках. Выбрать месяц, нажать кнопку ЗАПОЛНИТЬ.
8. Сформировать налоговую отчётность: декларацию по налогу на имущество.
9. ОТЧЁТЫ-РЕГЛАМЕНТИРОВАННЫЕ: - НАЛОГОВАЯ ОТЧЁТНОСТЬ-ИМУЩЕСТВО. Расчёт общей суммы налога - выбрать месяц, нажать кнопку ЗАПОЛНИТЬ