

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 19.04.2022 05:47:24  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d426d39e5f1c1feabb73e743d14a4831da36d089

**МИНОБРАЗОВАНИЯ РОССИИ**  
**Федеральное государственное бюджетное образовательное учреждение высшего образования**  
**«Юго-Западный государственный университет»**  
**(ЮЗГУ)**

Кафедра вычислительной техники



## **ОПИСАНИЕ И АНАЛИЗ ПРОЕКТА ИС**

Методические указания по выполнению  
практических работ по дисциплине  
«Проектный практикум»  
для обучающихся по направлению подготовки  
09.03.02 Информационные системы направленность  
(профиль) "Информационные технологии в бизнесе"

Курск 2022

УДК 004.82 (075.8)

Составитель Т.И.Лапина

Рецензент

Кандидат технических наук, доцент Е.А.Петрик

**Описание и анализ проекта ИС:** методические указания по выполнению практических работ по дисциплине «Проектный практикум» / Юго-Зап. гос. ун-т; сост.: Т. И. Лапина, Курск, 2022. 64с. ил. 44, табл.4, Билиогр.: с.64.

Содержат краткие теоретические сведения о методах разработки требований к проекту информационных систем.

09.03.02 Методические указания соответствуют требованиям программ по направлениям подготовки бакалавров: 09.03.02 Информационные системы, направленность (профиль) Информационные технологии в бизнесе

Предназначены для студентов направления подготовки бакалавров 09.03.02 Информационные системы направленность (профиль) "Информационные технологии в бизнесе" дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать 17.01.22 Формат 60x84 1/16.

Усл. печ. л.0,9 . Уч. – изд. л.0.8 .Тираж 100 экз. Заказ. 170

Бесплатно.

Юго - Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

# **Практическая работа №1**

## **Анализ требований при проектировании ИС**

### **1. Цель работы**

Получить навыки описания и анализа требований к проектируемой информационной системе, методы организации коллектива и распределения ролей в группе разработчиков.

### **2. Основные теоретические положения**

Практическая работа направлена на ознакомление с процессом описания информационной системы и получение навыков по использованию основных методов анализа ИС.

Требования к результатам выполнения практикума:

1. наличие описания информационной системы;
2. проведение анализа осуществимости выполнения проекта;
3. наличие заключения о возможности реализации проекта, содержащего рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.

#### **2.1 Общие сведения о разработке ИС**

Проблемы управления программными проектами впервые проявились в 60-х - начале 70-х годов, когда провалились многие большие проекты по разработке программных продуктов. Были зафиксированы задержки в создании ПО, оно было ненадежным, затраты на разработку в несколько раз превосходили первоначальные оценки, созданные про-

граммные системы часто имели низкие показатели производительности. Причины провалов коренились в тех подходах, которые использовались в управлении проектами. Применяемая методика была основана на опыте управления техническими проектами и оказалась неэффективной при разработке программного обеспечения.

Важно понимать разницу между профессиональной разработкой ПО и любительским программированием. Необходимость управления программными проектами вытекает из того факта, что процесс создания профессионального ПО всегда является субъектом бюджетной политики организации, где оно разрабатывается, и имеет временные ограничения. Работа *руководителя программного проекта* по большому счету заключается в том, чтобы гарантировать выполнение этих бюджетных и временных ограничений с учетом бизнес-целей организации относительно разрабатываемого ПО.

*Руководители проектов* призваны спланировать все этапы разработки программного продукта. Они также должны контролировать ход выполнения работ и соблюдения всех требуемых стандартов. Постоянный контроль за ходом выполнения работ необходим для того, чтобы процесс разработки не выходил за временные и бюджетные ограничения. Хорошее управление не гарантирует успешного завершения проекта, но плохое управление обязательно приведет к его провалу. Это может выразиться в задержке сроков сдачи готового ПО, в превышении сметной стоимости проекта и в несоответствии готового ПО спецификации требований.

Процесс разработки ПО существенно отличается от процессов реализации технических проектов, что порождает определенные сложности в управлении программными проектами:

1. *Программный продукт нематериален.* Программное обеспечение нематериально, его нельзя увидеть или потрогать. Руководитель программного проекта не видит процесс "роста" разрабатываемого ПО. Он может полагаться только на документацию, которая фиксирует процесс разработки программного продукта.

2. *Не существует стандартных процессов разработки ПО.* На сегодняшний день не существует четкой зависимости между процессом создания ПО и типом создаваемого программного продукта. Другие технические дисциплины имеют длительную историю, процессы разработки технических изделий многократно опробованы и проверены. Процессы создания большинства технических систем хорошо изучены. Изучением же процессов создания ПО специалисты занимаются только последнее время. Поэтому пока нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему программному проекту.

3. *Большие программные проекты - это часто "одноразовые" проекты.* Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, руководители проектов должны обладать очень большим практическим опытом. Но постоянные технологические изменения в компьютерной технике и коммуникационном оборудовании обесценивают предыдущий опыт. Знания

и навыки, накопленные опытом, могут не востребоваться в новом проекте.

Перечисленные отличия могут привести к тому, что реализация проекта выйдет из временного графика или превысит бюджетные ассигнования. Программные системы зачастую оказываются новинками как в "идеологическом", так и в техническом плане. Поэтому, предвидя возможные проблемы в реализации программного проекта, следует всегда помнить, что многим из них свойственно выходить за рамки временных и бюджетных ограничений.

## **2.2 Процесс управления разработкой программного обеспечения**

Невозможно описать и стандартизировать все работы, выполняемые в проекте по созданию ПО. Эти работы весьма существенно зависят от организации, где выполняется разработка ПО, и от типа создаваемого программного продукта. Но всегда можно выделить следующие:

- Написание предложений по созданию ИС.
- Планирование и составление графика работ по созданию ИС.
- Оценивание стоимости проекта ИС.
- Контроль за ходом выполнения работ.

Первая стадия программного проекта может состоять из написания предложений по реализации этого проекта. Предложения должны содержать описание целей проектов и способов их достижения. Они также обычно включают в себя оценки финансовых и временных за-

трат на выполнение проекта. При необходимости здесь могут приводиться обоснования для передачи проекта на выполнение сторонней организации или команде разработчиков.

*Написание предложений* — очень ответственная работа, так как для многих организаций вопрос о том, будет ли проект выполняться самой организацией или разрабатываться по контракту сторонней компанией, является критическим. Не существует каких-либо рекомендаций по написанию предложений, многое здесь зависит от опыта.

На этапе *планирования проекта* определяются процессы, этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта. Реализация этого плана приведет к достижению целей проекта. Определение стоимости проекта напрямую связано с его планированием, поскольку здесь оцениваются ресурсы, требующиеся для выполнения плана.

*Контроль за ходом выполнения работ (мониторинг проекта)* — это непрерывный процесс, продолжающийся в течение всего срока реализации проекта. Руководитель должен постоянно отслеживать ход реализации проекта и сравнивать фактические и плановые показатели выполнения работ с их стоимостью. Хотя многие организации имеют механизмы формального мониторинга работ, опытный руководитель может составить ясную картину о стадии развития проекта просто путем неформального общения с разработчиками.

Неформальный мониторинг часто помогает обнаружить потенциальные проблемы, которые в явном виде могут обнаружиться позднее. Например, ежедневное обсуждение хода выполнения работ может выявить отдельные недоработки в создаваемом программном продукте.

Вместо ожидания отчетов, в которых будет отражен факт "пробуксовки" графика работ, можно обсудить со специалистами намечающиеся программистские проблемы и не допустить срыва графика работ.

В течение реализации проекта обычно происходит несколько формальных контрольных проверок хода выполнения работ по созданию ПО. Такие проверки должны дать общую картину хода реализации проекта в целом и показать, насколько уже разработанная часть ПО соответствует целям проекта.

Время выполнения больших программных проектов может занимать несколько лет. В течение этого времени цели и намерения организации, заказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным либо исходные требования к создаваемому ПО просто устарели и их необходимо кардинально менять. В такой ситуации руководство организации-разработчика может принять решение о прекращении разработки ПО или об изменении проекта в целом с тем, чтобы учесть изменившиеся цели и намерения организации-заказчика.

Руководители проектов обычно обязаны сами *подбирать исполнителей* для своих проектов. В идеальном случае профессиональный уровень исполнителей должен со-ответствовать той работе, которую они будут выполнять в ходе реализации проекта.

Однако во многих случаях руководители должны полагаться на команду разработчиков, которая далека от идеальной. Такая ситуация может быть вызвана следующими причинами:



1. Бюджет проекта не позволяет привлечь высококвалифицированный персонал. В таком случае за меньшую плату привлекаются менее квалифицированные специалисты.

2. Бывают ситуации, когда невозможно найти специалистов необходимой квалификации как в самой организации-разработчике, так и вне ее. Например, в организации "лучшие люди" могут быть уже заняты в других проектах.

3. Организация хочет повысить профессиональный уровень своих работников. В этом случае она может привлечь к участию в проекте неопытных или недостаточно квалифицированных работников, чтобы они приобрели необходимый опыт и по-учились у более опытных специалистов.

Таким образом, почти всегда подбор специалистов для выполнения проекта имеет определенные ограничения и не является свободным. Вместе с тем необходимо, чтобы хотя бы несколько членов группы разработчиков имели квалификацию и опыт, достаточные для работы над данным проектом. В противном случае невозможно избежать ошибок в разработке ПО.

Руководитель проекта обычно обязан посылать *отчеты* о ходе его выполнения как заказчику, так и подрядным организациям. Это должны быть краткие документы, основанные на информации, извлекаемой из подробных отчетов о проекте. В этих отчетах должна быть та информация, которая позволяет четко оценить степень готовности создаваемого программного продукта.

В рамках курса «Технология разработки программного обеспечения» выделены следующие роли в группе по разработке ПО:

- Руководитель - общее руководство проектом, написание документации, общение с заказчиком ПО
- Системный аналитик - разработка требований (составление технического задания, проекта программного обеспечения)
- Тестер - составление плана тестирования и аттестации готового ПО (продукта), составление сценария тестирования, базовый пример, проведение мероприятий по плану тестирования
- Разработчик - моделирование компонент программного обеспечения, кодирование

#### Планирование проекта разработки программного обеспечения

Эффективное управление программным проектом напрямую зависит от правильного планирования работ, необходимых для его выполнения. План помогает руководителю предвидеть проблемы, которые могут возникнуть на каких-либо этапах создания ПО, и разработать превентивные меры для их предупреждения или решения. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

Процесс планирования начинается, исходя из описания системы, с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием про-

ектных параметров, таких как структура и размер проекта, а также распределением функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты документация, прототипы, подсистемы или версии программного продукта) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. После этого проводится контроль выполнения работ и отмечаются расхождения между реальным и плановым ходом работ.

Далее, по мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены (и согласованы с заказчиком ПО) проектные ограничения.

Конечно, большинство руководителей проектов не думают, что реализация их проектов пройдет гладко, без всяких проблем. Желательно описать возможные проблемы еще до того, как они проявят себя в ходе выполнения проекта. Поэтому лучше составлять "пессимистические" графики работ, чем "оптимистические". Но, конечно, невозможно построить план, учитывающий все, в том числе случайные, проблемы и задержки выполнения проекта, поэтому и возникает необходимость периодического пересмотра проектных ограничений и этапов создания программного продукта.

План проекта должен четко показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график

выполнения этих этапов. В некоторых организациях план проекта составляется как единый документ, содержащий все виды планов, описанных выше. В других случаях план проекта описывает только технологический процесс создания ПО. В таком плане обязательно присутствуют ссылки на планы других видов, но они разрабатываются отдельно от плана проекта.

Детализация планов проектов очень различается в зависимости от типа разрабатываемого программного продукта и организации-разработчика. Но в любом случае большинство планов содержат следующие разделы.

1. *Введение.* Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом.

2. *Организация выполнения проекта.* Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.

3. *Анализ рисков.* Описание возможных проектных рисков, вероятности их проявления и стратегий, направленных на их уменьшение.

4. *Аппаратные и программные ресурсы, необходимые для реализации проекта.* Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта. Если аппаратные средства требуется закупать, приводится их стоимость совместно с графиком закупки и поставки.

5. *Разбиение работ на этапы.* Процесс реализации проекта разбивается на отдельные процессы, определяются этапы выполнения

проекта, приводится описание результатов ("выходов") каждого этапа и контрольные отметки.

1. *График работ.* В этом графике отображаются зависимости между отдельными процессами (этапами) разработки ПО, оценки времени их выполнения и распределение членов команды разработчиков по отдельным этапам.

2. *Механизмы мониторинга и контроля за ходом выполнения проекта.*

Описываются предоставляемые руководителем отчеты о ходе выполнения работ, сроки их предоставления, а также механизмы мониторинга всего проекта.

План должен регулярно пересматриваться в процессе реализации проекта. Одни части плана, например график работ, изменяются часто, другие более стабильны. Для внесения изменений в план требуется специальная организация документопотока, позволяющая отслеживать эти изменения.

### **2.3. Общие сведения о требованиях к информационным системам**

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, назы-

вается требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений - разработкой требований.

Требования подразделяются на пользовательские и системные. Пользовательские требования - это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования - это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

#### **2.4. Первые шаги по разработке требований к информационным системам - анализ осуществимости**

Разработка требований - это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Для новых программных систем процесс разработки требований должен начинаться с анализа осуществимости. Началом такого анализа является общее описание системы и ее назначения, а результатом анализа — отчет, в котором должна быть четкая рекомендация, продолжать или нет процесс разработки требований проектируемой системы. Другими словами, анализ осуществимости должен осветить следующие вопросы.

Отвечает ли система общим и бизнес-целям организации-заказчика и организации-разработчика?

Можно ли реализовать систему, используя существующие на данный момент технологии и не выходя за пределы заданной стоимости?

Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Критическим является вопрос, будет ли система соответствовать целям организации. Если система не соответствует этим целям, она не представляет никакой ценности для организации. В то же время многие организации разрабатывают системы, не соответствующие их целям, либо не совсем ясно понимая эти цели, либо под влиянием политических или общественных факторов.

Выполнение анализа осуществимости включает сбор и анализ информации о будущей системе и написание соответствующего отчета. Сначала следует определить, какая именно информация необходима, чтобы ответить на поставленные выше вопросы. Например, эту информацию можно получить, ответив на следующее:

1. Что произойдет с организацией, если система не будет введена в эксплуатацию?
2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?
3. Каким образом система будет способствовать целям бизнеса?
4. Требуется ли разработка системы технологии, которая до этого не использовалась в организации?

Далее необходимо определить источники информации. Это могут быть менеджеры отделов, где система будет использоваться, разработчики программного обеспечения, знакомые с типом будущей системы, технологи, конечные пользователи и т.д.

После обработки собранной информации готовится отчет по анализу осуществимости создания системы. В нем должны быть даны рекомендации относительно продолжения разработки системы. Могут быть предложены изменения бюджета и графика работ по созданию системы или предъявлены более высокие требования к системе.

### **Порядок выполнения работы**

1. Изучить предлагаемый теоретический материал.
- .2. Составить подробное описание информационной системы.
- .3. На основании описания системы провести анализ осуществимости.

В ходе анализа ответить на вопросы

1.Что произойдет с организацией, если система не будет введена в эксплуатацию ?

2.Какие текущие проблемы существуют в организации и как новая система поможет их решить?

Каким образом система будет способствовать целям бизнеса? Требуется ли разработка системы технологии, которая до этого не использовалась в организации?



Результатом анализа должно явиться заключение о возможности реализации проекта.

4. Распределить роли в группе (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик).

Заполнить разделы плана:

1. *Введение*

2. *Организация выполнения проекта*

3. *Анализ рисков*

Разделы должны содержать рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.

Составить отчет о проделанной работе.

### **3. Содержание отчета**

В отчете следует указать:

1. Цель работы

2. Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом

3. Описание информационной системы (ПО)

наличие заключения о возможности реализации проекта, содержащего рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению

4. Анализ осуществимости (согласно требованиям к результатам выполнения лабораторного практикума п.2), указать возможные проблемы и пути их решения.

5. Роли участников группы разработки ПО.

6. Программно-аппаратные средства, используемые при выполнении работы.

7. Заключение (выводы)

8. Список используемой литературы

#### **4. Контрольные вопросы**

1. Какие вопросы включает методика описания проектируемой ИС?

2. На каких уровнях проводится обследование аспектов деятельности предприятий?

3. Какие существуют универсальные методы, пригодные для обследования всех функциональных звеньев предприятия?

4. Какие существуют документы для описания ИС?

5. Что такое «реинжиниринг» объекта автоматизации?

6. С какой целью проводится «реинжиниринг» объекта автоматизации?

7. В каких направлениях выполняется информационный анализ предметной области?

8. Как систематизируется полученная информация?

9. Кто выполняет функции анализа объекта автоматизации?

10. Цель анализа полученной информации.

#### **Список литературы**

1. Лапина, Т. И. Информационные системы. Проектный практикум к выполнению и защите ВКР бакалавра по направлению 09.03.02 Информационные системы, 09.03.03 Прикладная информатика/ Т. И. Лапина//Юго-Западный гос. ун-т, ЗАО «Университетская книга»—Курск, 2016.—99с.

2. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс] : учебное пособие / С. Ю. Золотов. - Томск : Эль Контент, 2013. - 88 с.

3. Абрамов, Г. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / Г. В. Абрамов, И. Медведкова, Л. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с.

4. Стасышин, В. М. Проектирование информационных систем и баз данных [Электронный ресурс] : учебное пособие / В. М. Стасышин. - Новосибирск : НГТУ, 2012. - 100 с.

5. Вендров, А.М. Проектирование программного обеспечения [Текст] : учебник / А.М. Вендров.— М: Финансы и Статистика, 2006. — 352с.

## Практическая работа №2

### Постановка задачи проектирования ИС. Модель Захмана.

#### 1. Цель работы

Получение навыков анализа объекта автоматизации, выявления целей задач проектируемой ИС, разработка описательных моделей для согласования интересов заказчика и разработчиков

#### 2 Основные теоретические положения

В 1987 году *Джон Захман* опубликовал схему анализа информационной системы. Модель Захмана преследует *две основные цели*:

- с одной стороны, логически разбить все описание архитектуры на отдельные разделы для упрощения их формирования и восприятия,
- с другой обеспечить возможность рассмотрения целостной архитектуры с выделенных точек зрения или соответствующих уровней абстракции.

Основная идея заключается в том, чтобы обеспечить возможность последовательного описания каждого отдельного аспекта системы *в координации со всеми остальными*.

Собственно модель представляется в виде таблицы, имеющей пять строк и шесть столбцов, которая приведена на рис. 1.

В модели пять строк, отображенная на рисунке шестая строка соответствует уже не уровню описания архитектуры, а уровню работающей системы или предприятия в целом.

*Перспективы* (строки в таблице) соответствуют различным точкам зрения специалистов различных уровней управления предприятием при разработке информационной системы.

Две верхние строки соответствуют наиболее общим представлениям руководителей компаний – заказчиков ИС, планы и цели.

"*Контекст*" соответствует уровню интересов высшего руководства и собрания акционеров. Второй уровень "*Модель предприятия*" соответствует интересам бизнес-менеджеров и владельцев процессов.

		Данные ЧТО	Функции КАК	Дислокация, сеть ГДЕ	Люди КТО	Время КОГДА	Мотивация ПОЧЕМУ	
Бизнес-руководители	Планировщик	Список важных понятий и объектов	Список основных бизнес-процессов	Территориальное расположение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
	Владелец, менеджер	Концептуальная модель данных	Модель бизнес-процессов	Схема логистики	Модель потока работ (workflow)	Мастер-план реализации	Бизнес-план	Модель предприятия
	Конструктор, архитектор	Логические модели данных	Архитектура приложений	Модель распределенной архитектуры	Архитектура интерфейса пользователя	Структура процессов	Роли и модели бизнес-правил	Модель системы
IT-менеджеры и разработчики	Проектировщик	Физическая модель данных	Системный проект	Технологич. архитектура	Архитектура презентации	Структуры управления	Описания бизнес-правил	Технологическая (физическая) модель
	Разработчик	Описание структуры данных	Программный код	Сетевая архитектура	Архитектура безопасности	Определение временных привязок	Реализация бизнес-логики	Детали реализации
		Данные	Работающие программы	Сеть	Реальные люди, организации	Бизнес-события	Работающие бизнес-стратегии	Работающее предприятие
		Данные	Функции, Процессы	Сеть, расположение систем	Люди, организации	Время, расписания	Мотивация	

Рисунок 1 – Модель Захмана

Третий уровень "*Модель системы (Логический уровень)*" – это уровень, на котором бизнес-менеджеры, бизнес-аналитики и менеджеры, отвечающие за ИТ, должны работать вместе.

Уровни с четвертого и далее описывают детали, которые представляют интерес для ИТ-менеджеров, проектировщиков, разработчиков.

На каждом из этих уровней участники рассматривают одни и те же категории вопросов, соответствующих *столбцам* в таблице, – только с различным уровнем абстракции и детализации.

В содержание этих колонок входят:

- используемые данные (что?);
- процессы и функции (как?);
- места выполнения этих процессов (где?);
- организации, персоналии-участники, системы (кто?);

- управляющие события (когда?);
- цели и ограничения, определяющие работу системы (зачем?).

*Основные правила заполнения таблицы следующие:*

- каждая клетка таблицы независима от других, вместе они образуют функционально полное описание системы ("базис");
- порядок следования колонок несущественен;
- каждая клетка содержит соответствующее описание аспекта реализации системы в виде определенной модели или простого описания (текстового документа);
- базовые модели для каждой из колонок являются уникальными;
- соответствующие модели в клетках каждого ряда в совокупности образуют полное описание системы с выбранной перспективы;
- заполнение клеток должно проводиться последовательно "сверху вниз", попытка пропуска одного из рядов является, скорее, "шаманством" (в том плане, что нельзя создать хорошо работающую систему, "перепрыгнув" определенные уровни ее описания на этапе проектирования).

Рассмотрим *заполнение строк*.

*Первая строка* соответствует уровню планирования бизнеса в целом (*бизнес-модель*). На этом уровне вводятся достаточно общие основные понятия, определяющие бизнес – например, продукты и услуги, клиенты, расположение объектов бизнеса, а также формулируется бизнес-стратегия (колонок 6 – "Мотивация"). Фактически, данная строка определяет контекст всех последующих строк.

*Вторая строка (концептуальная модель)* предназначена для определения в терминах бизнеса структуры организации, ключевых и вспомогательных бизнес-процессов.

*Третий уровень (логическая модель)* соответствует рассмотрению с точки зрения Системного Архитектора. Здесь бизнес-процессы описываются уже в терминах информационных систем, включая различные типы данных, правила их преобразования и обработки для выполнения определенных на уровне 2 бизнес-функций.

На *четвертом уровне – технологической или физической модели* – осуществляется привязка данных и операций над ними к выбранным

технологиям реализации. Например, здесь может быть определен выбор реляционной СУБД, или средств работы с неструктурированными данными, или объектно-ориентированной среды.

*Пятый уровень* соответствует *детальной реализации системы*, включая конкретные модели оборудования, топологию сети, производителя и версию СУБД, средства разработки и собственно готовый программный код. Многие из работ на данном уровне часто выполняются субподрядчиками.

Последний, *шестой уровень* описывает *работающую систему*. На этом уровне могут быть введены, в том числе, такие объекты, как инструкции для работы с системой, фактические базы данных. Надо заметить, что в исходной работе Захмана содержание этого уровня не детализируется. При развитии модели, как будет показано ниже, отмечены возможности рассмотрения аспектов функционирования работающей системы с точки зрения, например, конечного пользователя или эксплуатирующих служб.

Рассмотрим теперь, как осуществляется последовательная детализация отдельных аспектов описания системы, для чего обратим внимание на различные *колонки таблицы*. Напомним, что порядок расположения колонок в таблице, вообще говоря, произволен.

*Первая колонка* отвечает на вопрос "*ЧТО?*" и определяет используемые в системе *данные*. На *верхнем уровне* достаточным будет простое перечисление основных объектов, используемых в бизнесе. На *втором уровне* данные объекты объединяются в семантическую модель высокого уровня и обычно описываются в виде *диаграммы "сущности-связи"* (Е-Р диаграммы) с отражением основных связей и наиболее существенных бизнес-ограничений. На *третьем уровне* эта модель приводится к нормализованной форме, определяются все атрибуты и ключи. *Четвертый уровень* представляет собой физическую модель данных в системе (в объектно-ориентированном подходе – иерархию классов). *Пятый уровень* содержит описание модели на языке управления данными для формирования таблиц, готовые библиотеки классов, табличные пространства СУБД. Наконец, *последний уровень* может описывать фактические наборы данных, в том числе такие характеристики, как журналы доступа, размеры реально занимаемого дискового пространства, статистику обращений и т. п. Конечно, можно отметить определенное *несовершенство данной модели при использова-*

*нии объектно-ориентированного подхода* – фактически модель предписывает раздельное рассмотрение данных (свойств) и функций (методов) классов.

Колонка *функций* (ответ на вопрос "КАК?") предназначена для последовательной детализации описания того, как миссия предприятия реализуется на уровне отдельных операций. В частности, на *первом уровне* достаточным будет простое перечисление бизнес-процессов. *Второй уровень* будет содержать модель бизнес-процессов, которая впоследствии детализируется в операции над данными и архитектуру приложений (*уровень 3*), методы классов (*уровень 4*), программный код (*уровень 5*) и, наконец, исполняемые модули. При этом, начиная с 4-го уровня, рассмотрение ведется уже не в рамках Предприятия в целом, а по отдельным подсистемам или приложениям.

Следующая колонка (вопрос "ГДЕ?") определяет *пространственное распределение компонент системы и сетевую организацию*. На уровне планирования бизнеса здесь достаточно определить расположение всех производственных объектов. На следующем уровне эти объекты объединяются в модель со связями, характеризующими взаимодействие между собой, – будь то обмен информацией или поставки товаров. На третьем уровне системной архитектуры осуществляется привязка компонент информационной системы к узлам сети. Четвертый уровень служит для определения физической реализации в терминах аппаратных платформ, системного программного обеспечения, а также средств промежуточного уровня (так называемое "middleware"), используемых для интеграции различных компонент информационной системы между собой. Типичным примером могут являться брокеры запросов или средства обмена сообщениями. На пятом уровне определяются используемые протоколы и спецификации каналов связи. Последний уровень описывает функционирование реализованной сети.

Колонка таблицы, отвечающая на вопрос "КТО?", определяет *участников процесса*. На уровне планирования бизнеса здесь представлен список подразделений предприятия и выполняемые ими функции. На следующем уровне приводится полная организационная диаграмма, а также могут быть определены общие требования к информационной безопасности. Далее последовательно определяются участники бизнес-процессов и их роли, требования к интерфейсам пользователя и правила доступа к отдельным объектам, физическая их реализация на уровне кода или операторов определения доступа к таблицам



в СУБД. Последний уровень описывает обученных пользователей системы.

*Пятая колонка* отвечает на вопрос "КОГДА?" и определяет *временные характеристики бизнес-процессов и работы системы*. Детализация осуществляется сверху вниз, начиная от календарного плана (уровень 1) и основных параметров, характеризующих выполнение бизнес-процессов, – например, требование ко времени оформления сделки (уровень 2). На третьем уровне определяются события, вызывающие изменение состояния информационных объектов и инициацию операций над ними. На следующем уровне эти события транслируются в программные вызовы (триггеры) или передаваемые сообщения. Пятый уровень определяет физическую реализацию обработки таких событий. Наконец, на 6-м уровне – фактическая история функционирования системы.

Последняя колонка ("ПОЧЕМУ?" или "ЗАЧЕМ?") служит для определения *мотивации* и задает порядок перехода от задач бизнеса к требованиям и элементам информационных систем. Исходной точкой является бизнес-стратегия, которая затем последовательно транслируется в бизнес-план, затем в правила и ограничения для реализации бизнес-процессов, а на уровне 4 – в соответствующие приложения, необходимые для включения в состав информационных систем и, в дальнейшем, в их физическую реализацию.

Важным принципом модели Захмана является необходимость *последовательного перехода при углублении детализации рассмотрения*. Пропуск отдельных элементов, например, прямой переход от описания модели бизнес-процесса к физической реализации системы требует "привлечения магии" и почти всегда приводит к неудаче. На практике это часто случается при попытке разработки ИС на основании только устного описания требований пользователя.

### **3. Задание на лабораторную работу**

Описать требования к проектируемой ИС, включая: стратегические цели и задачи предприятия, бизнес архитектуру предприятия, архитектуру приложений. В рамках описания необходимо собрать и документировать следующую информацию:

Стратегические цели и задачи предприятия.

Основные бизнес - процессы организации.

Организационную структуру.  
Продукты и услуги компании.  
Информационные системы, функционирующие на предприятии.  
Инфраструктуру, поддерживающую существующие ИС.  
Документировать представленные выше данные рекомендуется в виде моделей и описания к ним.

#### **4 Контрольные вопросы**

1. Зачем нужно выяснять стратегические цели и задачи предприятия?
2. Зачем следует определять цель функционирования предприятия при проектировании архитектуры ИС?
3. Каким образом следует учитывать имеющуюся инфраструктуру предприятия?
4. Какие вопросы рассматриваются при анализе колонки модели Захмана «Что»?
5. Какие вопросы рассматриваются при анализе колонки модели Захмана «Как»?
6. Какие вопросы рассматриваются при анализе колонки модели Захмана «Где»?
7. Какие вопросы рассматриваются при анализе колонки модели Захмана «Кто»?
8. Какие вопросы рассматриваются при анализе колонки модели Захмана «Когда»?
9. Что такое архитектура предприятия «как есть» и «как будет»?
10. Какие документы являются результатом детализированного анализа предприятия?

#### **Список литературы**

1. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем [Текст] : учебник для студ. вуз. / А. М. Вендров. - М. : Финансы и статистика, 2000. - 352 с.
2. Карпова, Т. С. Базы данных [Электронный ресурс] : модели, разработка, реализация / Т. С. Карпова. - Москва : Интернет-Университет Информационных Технологий, 2008. - 357 с.

3. Смирнова, Г. Н. Проектирование экономических информационных систем [Текст] : учебник / А. А. Сорокин, Ю. Ф. Тельнов. - М. : Финансы и статистика, 2003.
4. Романов, В. П. Проектирование экономических информационных систем. Методология и современные технологии [Текст] : учебное пособие / В. П. Романов, Н. З. Емельянова, Т. Л. Партыка ; Российская экономическая академия им. Г. В. Плеханова. - М. : Экзамен, 2005. - 256 с.
5. Проектирование информационных систем [Электронный ресурс] : методические указания по изучению дисциплины для студентов и бакалавров специальностей 351400 (0808062.62) «Прикладная информатика (в экономике)» / Юго-Зап. гос. ун-т ; сост. В. Н. Николаев. - Курск : ЮЗГУ, 2010. - 22 с

## Практическая работа №3

### Расчет себестоимости разработки IT- проекта

В диалоге /ModelProperties(вызывается из меню Model/ModelProperties) во вкладке ABCUnits установите единицы измерения денег и времени — рубли и часы.

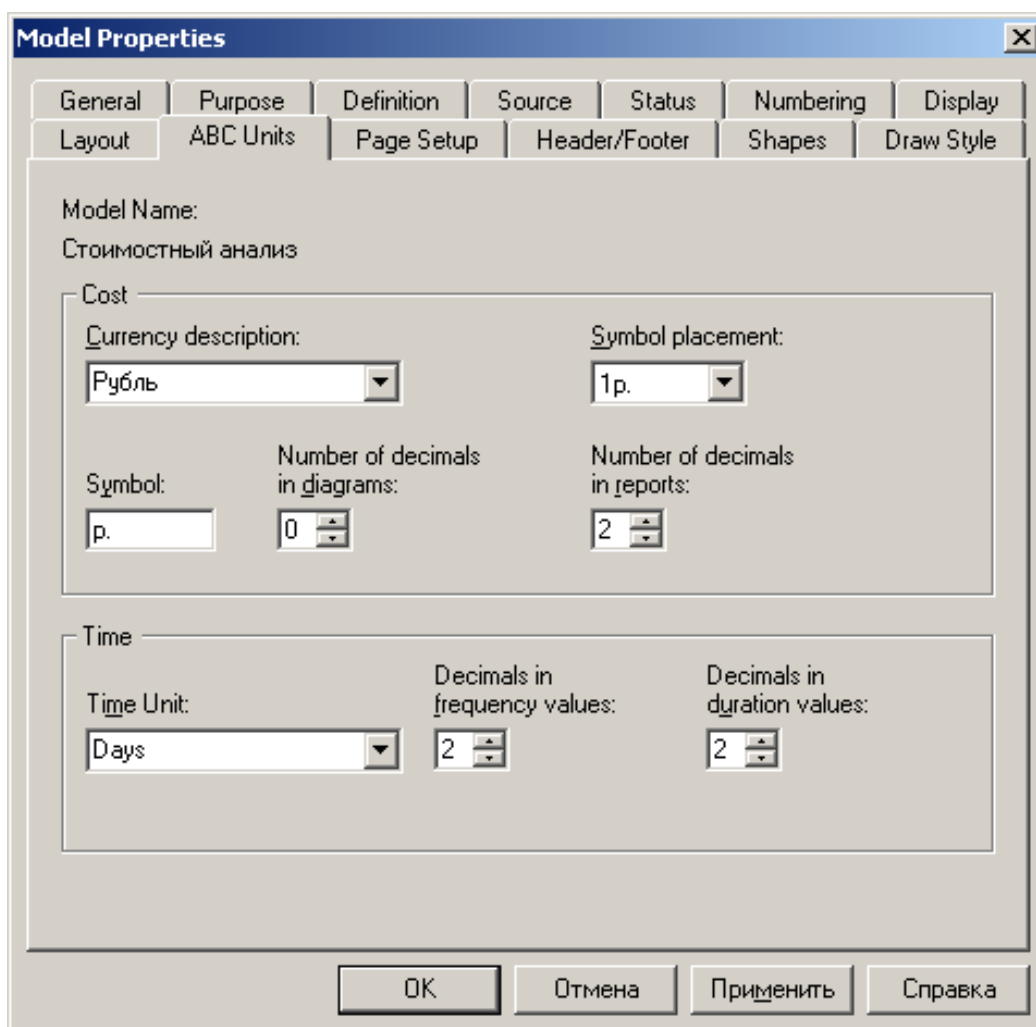


Рисунок 1 – Вкладка ABC Units диалога Model Properties

2. Перейдите в Dictionary/CostCenter и в диалоге CostCenterDictionary внесите название и определение центров затрат.

Таблица 1 – Определение центра затрат

Центр затрат	Определение
Управление	Затраты на управление, связанные с составлением графика работ, формированием партий компьютеров, контролем над сборкой и тестированием
Рабочая сила	Затраты на оплату рабочих, занятых сборкой и тестированием компьютеров
Компоненты	Затраты на закупку компонентов

Для отображения стоимости каждой работы в нижнем левом углу прямоугольника перейдите в меню Model/Model Properties и во вкладку Display диалого Model Properties включите опцию ABC Data (рис.2).

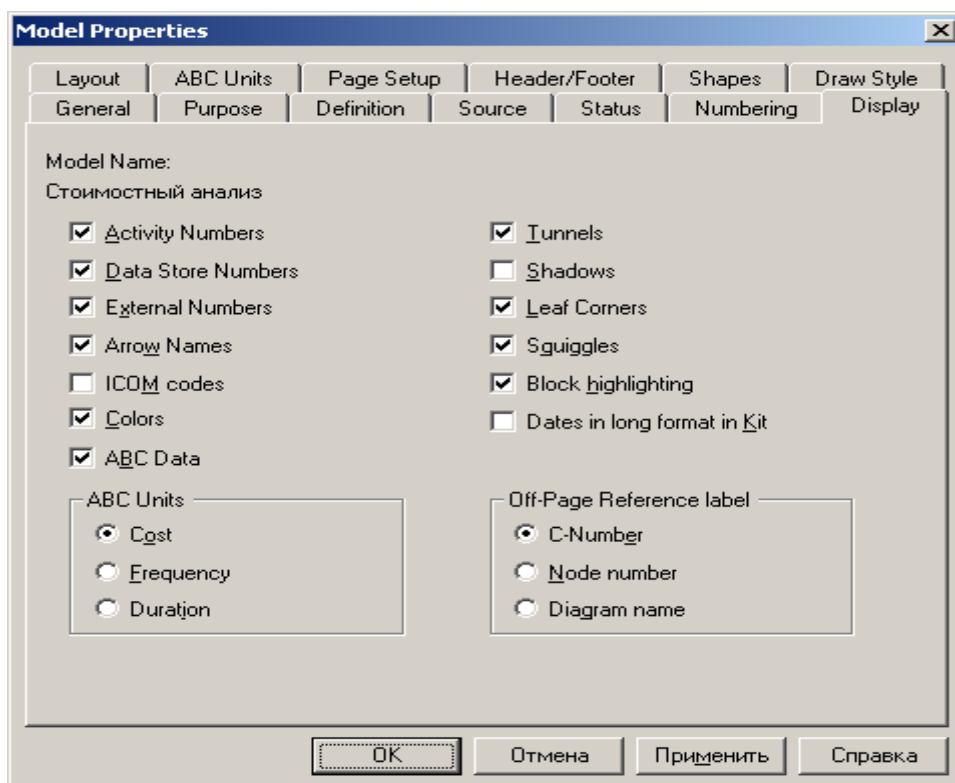


Рисунок 2 – Вкладка Display диалого Model Properties

Для отображения частоты или продолжительности работы переключите радиокнопки в группе ABCUnits.

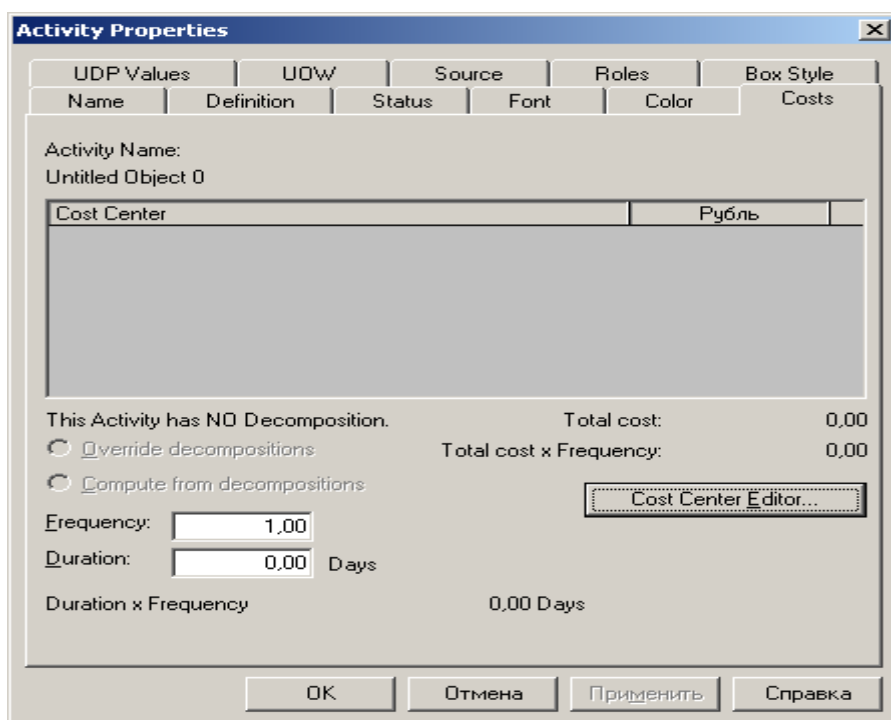


Рисунок 3 – Вкладка Cost диалога Activity Properties

Для назначения стоимости работе следует щелкнуть по ней правой кнопкой мыши и выбрать в контекстном меню Cost.

3. Для работ на диаграмме A2 внесите параметры ABC.

Таблица 2– Стоимости работ на диаграмме A2

Имя работы (Activity name)	Центр затрат (Cost Center)	Сумма центра затрат (Cost, руб.)	Продолжительность (Duration), день	Частота (Frequency)
Отслеживание расписания и управление сборкой и тестированием	Управление	500,00	1,00	1,00
Сборка настольных компьютеров	Рабочая сила	100,00	1,00	12,00
	Компоненты	16000,00		
		0		

Таблица 3 – Стоимости работ на диаграмме А2

Имя работы (Activity name)	Центр затрат (Cost Center)	Сумма центра затрат (Cost Center Cost), руб.	Продолжительность (Duration), день	Частота(Frequency)
Сборка ноутбуков	Рабочая сила	140,00	1,00	20,00
	Компоненты	28000,00		
Тестирование компьютеров	Рабочая сила	60,00	1,00	32,00

Посмотрите результат - стоимость работы верхнего уровня.

#### 4. Контрольные вопросы

1. Какие вопросы включает методика описания проектируемой ИС?
2. Какие существуют документы для описания ИС?
3. На каких уровнях проводится обследование стоимости разработки?
4. Как определяется сумма затрат на разработку проекта?
5. Какие существуют универсальные методы, пригодные для расчета стоимостных затрат?
6. Какие статьи входят в расчет стоимости затрат?
7. Какие инструментальные средства используются при расчете стоимости затрат?
8. Какие виды работ оцениваются при расчете стоимости затрат?
9. Оцениваются ли материальные ресурсы для реализации проекта?
10. Цель анализа полученной информации о стоимости разработки системы?

## Список литературы

1. Лапина, Т. И. Информационные системы. Проектный практикум к выполнению и защите ВКР бакалавра по направлению 09.03.02 Информационные системы, 09.03.03 Прикладная информатика/ Т. И. Лапина//Юго-Западный гос. ун-т, ЗАО «Университетская книга»—Курск, 2016.—99с.
2. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс] : учебное пособие / С. Ю. Золотов. - Томск : Эль Контент, 2013. - 88 с.
3. Абрамов, Г. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / Г. В. Абрамов, И. Медведкова, Л. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с.
4. Стасышин, В. М. Проектирование информационных систем и баз данных [Электронный ресурс] : учебное пособие / В. М. Стасышин. - Новосибирск : НГТУ, 2012. - 100 с.
5. Вендров, А.М. Проектирование программного обеспечения [Текст] : учебник / А.М. Вендров.— М: Финансы и Статистика, 2006. — 352с.



## Практическая работа №4

### Планирование разработки ИС. Диаграммы Ганта

#### 1. Цель работы

Разработать диаграмму Ганта выполнения проекта

#### 2. Порядок разработки диаграммы Ганта

Диаграммы Ганта – это способ графически изобразить ход проекта, порядок выполнения задач, их продолжительность, время начала и завершения. Рассмотрим на примере использование диаграммы Ганта.

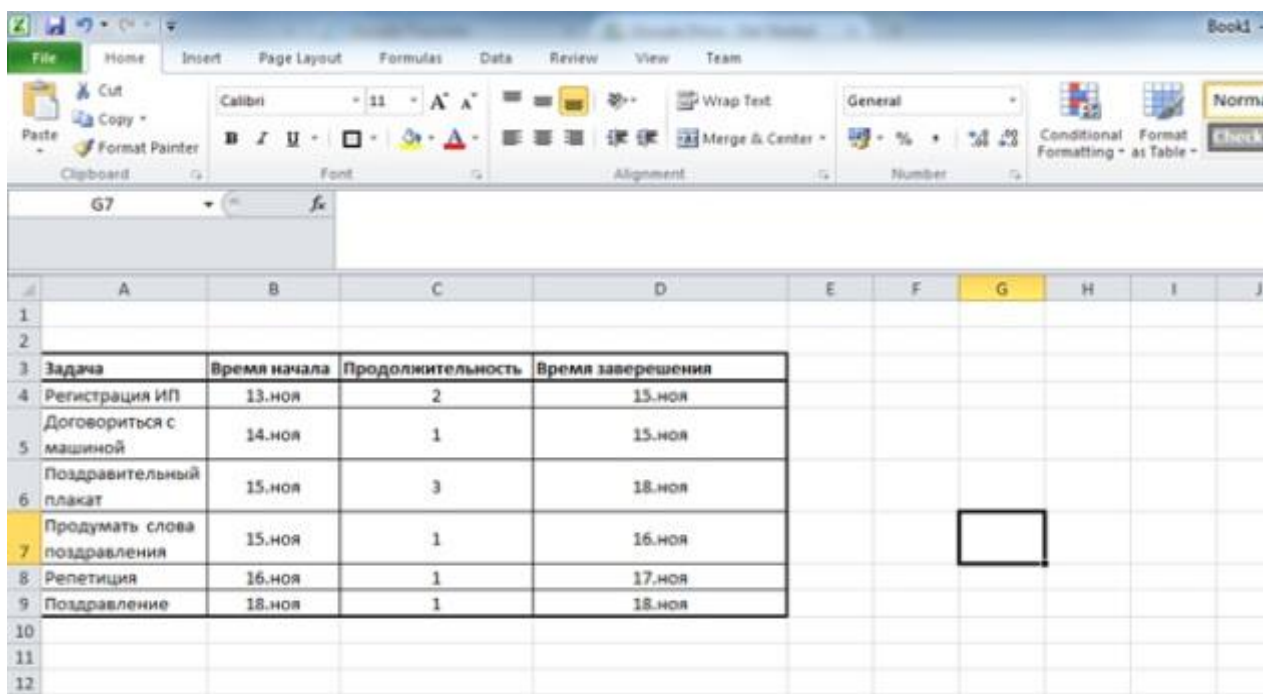
Таблица 4 – Перечень этапов выполнения проекта ИС

Этап проекта	Начало	Длительность	Конец
Изучение и анализ области внедрения	20.11.2016	3	22.11.2016
Составление технического задания	23.11.2016	4	26.11.2016
Планирование и контроль проектных работ.	Планирование и контроль проектных работ.	Планирование и контроль проектных работ.	Контроль проектных работ.
Описание входных и выходных данных	02.12.2016	7	08.12.2016
Разработка структур данных	09.12.2016	8	16.12.2016
Разработка технического проекта	17.12.2016	10	26.12.2016
Написание программ, модулей утилит	27.12.2016	10	05.01.2017
Отладка	06.01.2017	7	12.01.2017
Тестирование	13.01.2017	5	17.01.2017
Разработка справочной и технической документации	18.01.2017	2	19.01.2017
Внедрение	20.01.2017	7	26.01.2017
Итого		71	дней

Определим, как нам можно грамотно распланировать задачи с помощью диаграммы Ганта.

Воспользуемся инструментом Microsoft Excel. Построим диаграмму Ганта с помощью Microsoft Excel 2010.

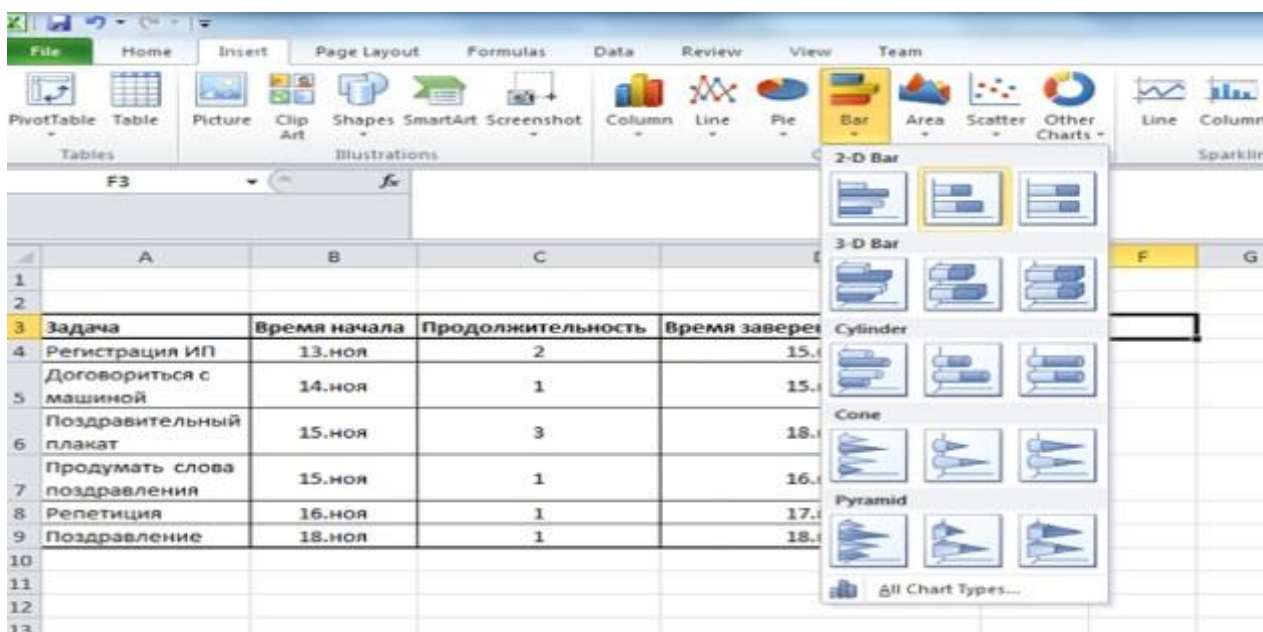
Открываем инструмент Excel и переносим наши задачи.



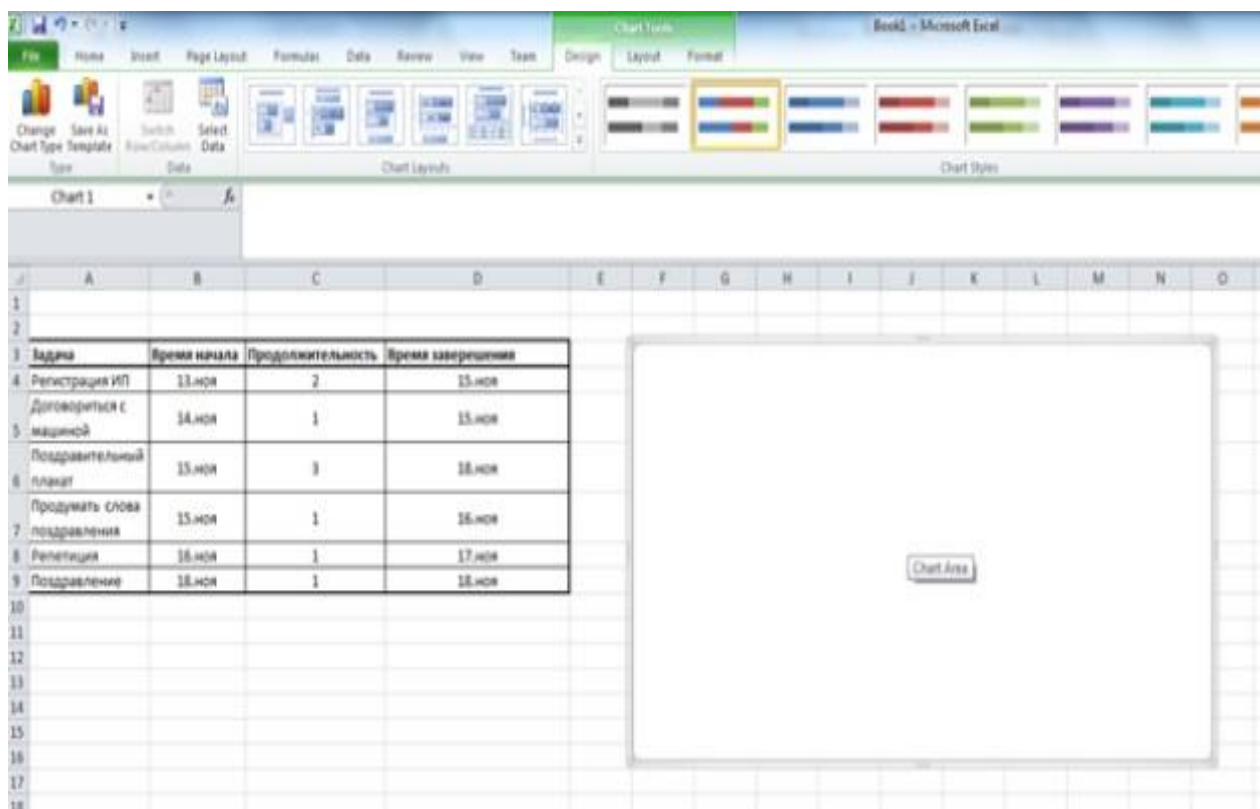
The screenshot shows the Microsoft Excel 2010 interface with a table of tasks. The table has four columns: 'Задача' (Task), 'Время начала' (Start Time), 'Продолжительность' (Duration), and 'Время завершения' (End Time). The data is as follows:

Задача	Время начала	Продолжительность	Время завершения
Регистрация ИП	13.ноя	2	15.ноя
Договориться с машиной	14.ноя	1	15.ноя
Поздравительный плакат	15.ноя	3	18.ноя
Продумать слова поздравления	15.ноя	1	16.ноя
Репетиция	16.ноя	1	17.ноя
Поздравление	18.ноя	1	18.ноя

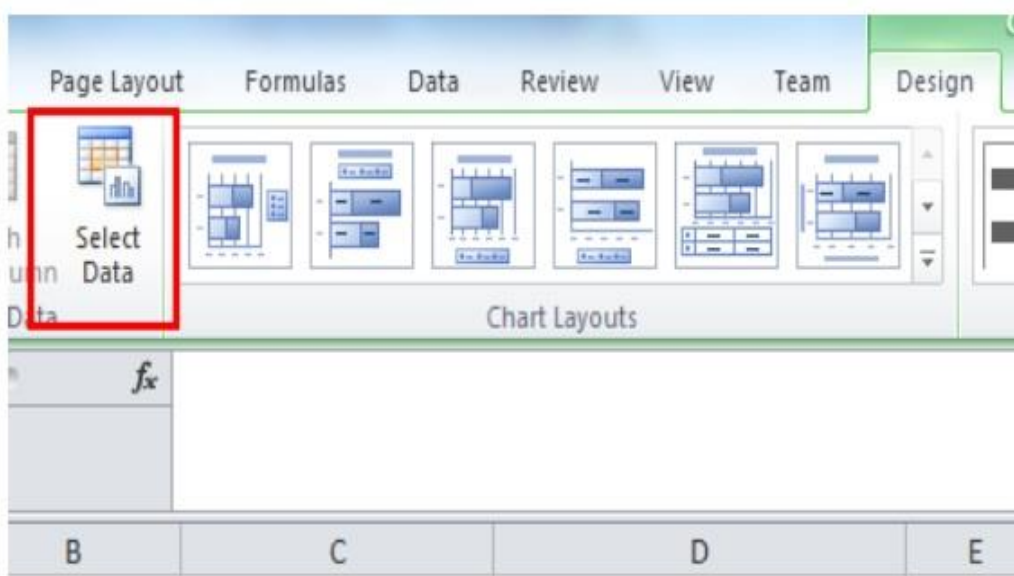
Следующая вещь, которую нам нужно сделать – это выбрать Insert -> Bar. И выбрать тот список диаграмм, который мы хотим отобразить



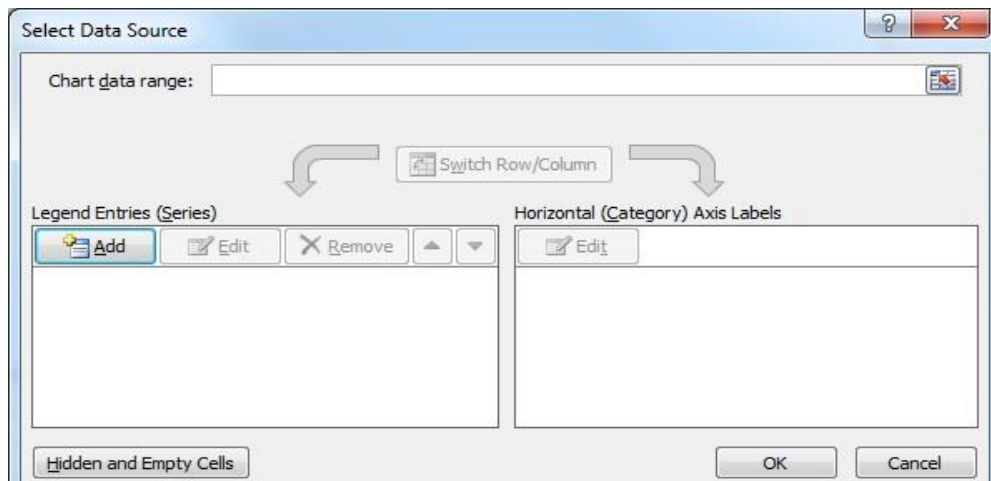
Появившуюся область мы отображаем в том месте, где хотим, чтобы наш график отображался. И теперь нам нужно связать эту область с нашими данными.



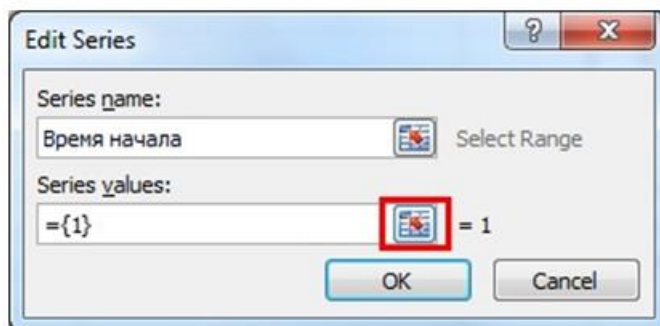
Далее на вкладке Design нажимаем кнопку Select Data.



В появившемся окне нажимает кнопку Add.

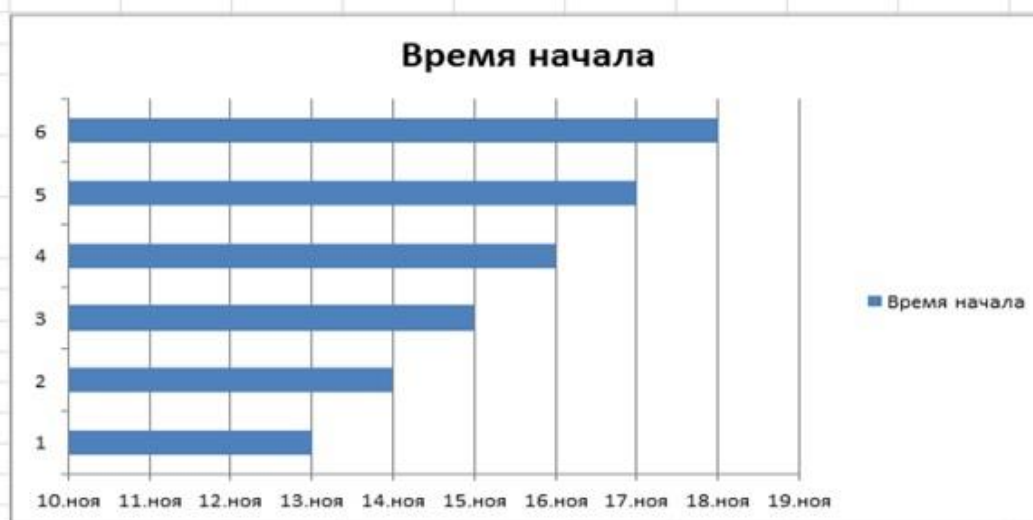


В появившемся окне, там где series name вводим Время начала и нажимаем на кнопку выделенную красным.

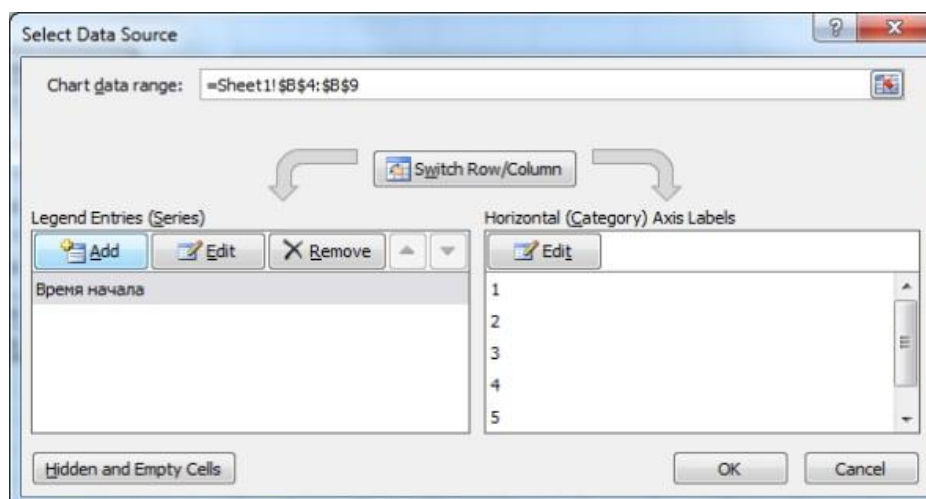


Когда появится окно с одним полем нажимаем на поле в окне Edit Series, после чего выделяем промежуток из таблицы, нужный нам и нажимаем на кнопку, выделенную красным.

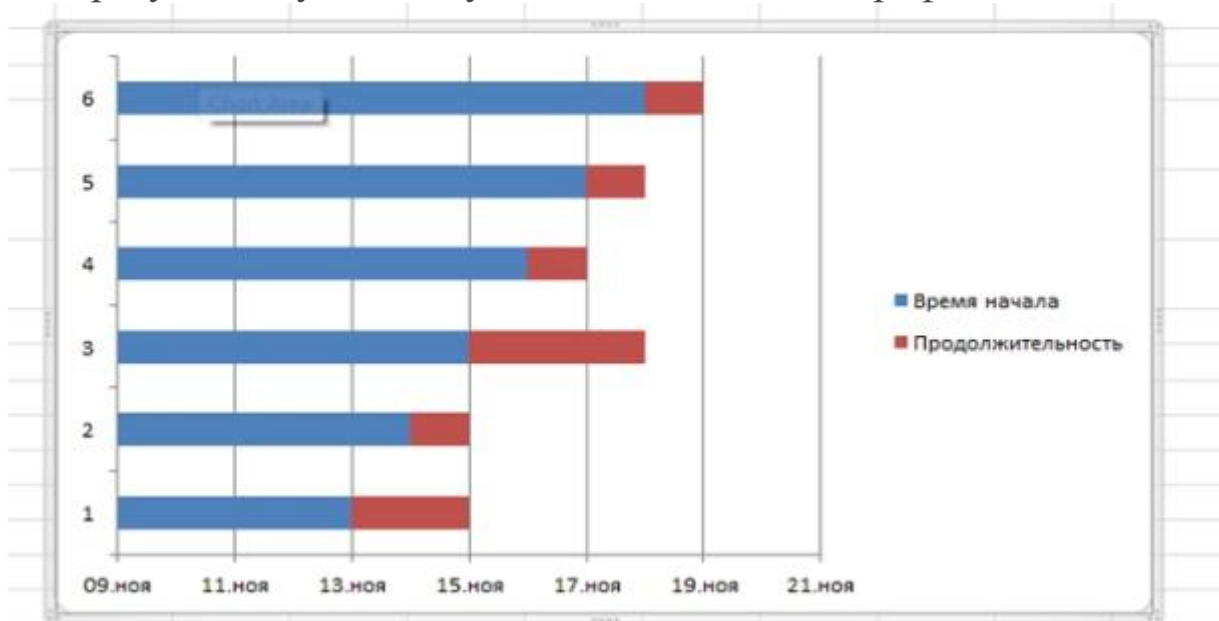
Получился следующий график



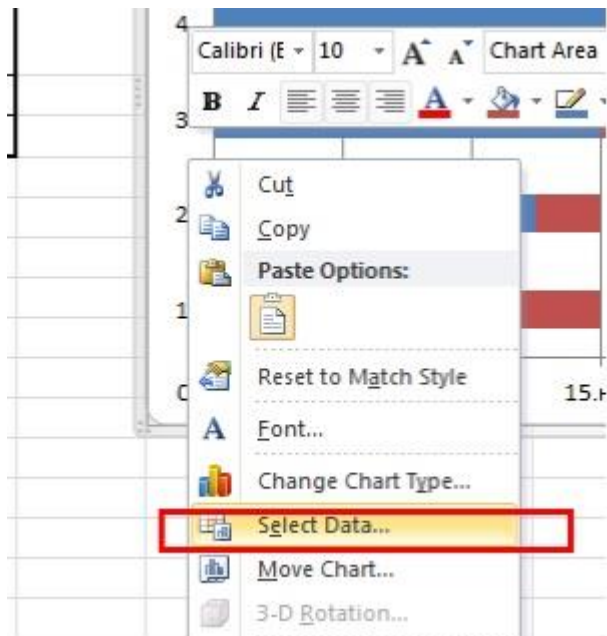
Необходимо в график добавить данные по продолжительности. Для этого в появившемся окне снова нажимаем кнопку Add и выбрать значения из графы продолжительность.



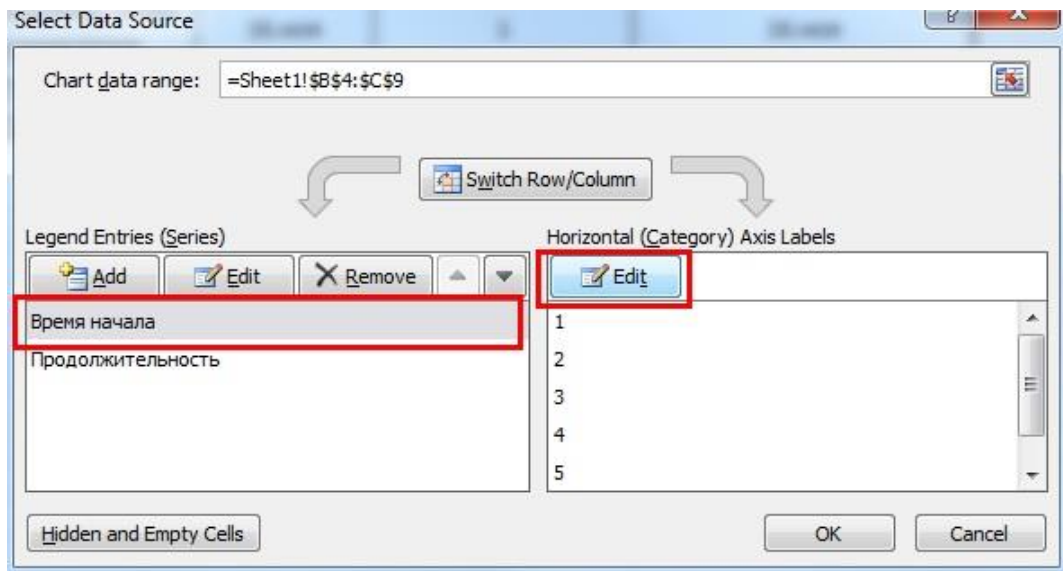
В результате у нас получился вот такой вот график.



Следующее, что мы сделаем – это заменим цифры, напротив колонок на названия из нашей таблицы (Столбик “Задача”). Для этого я щелкаю правой кнопкой мыши по цифрам и в появившемся меню выбираю Select Data.

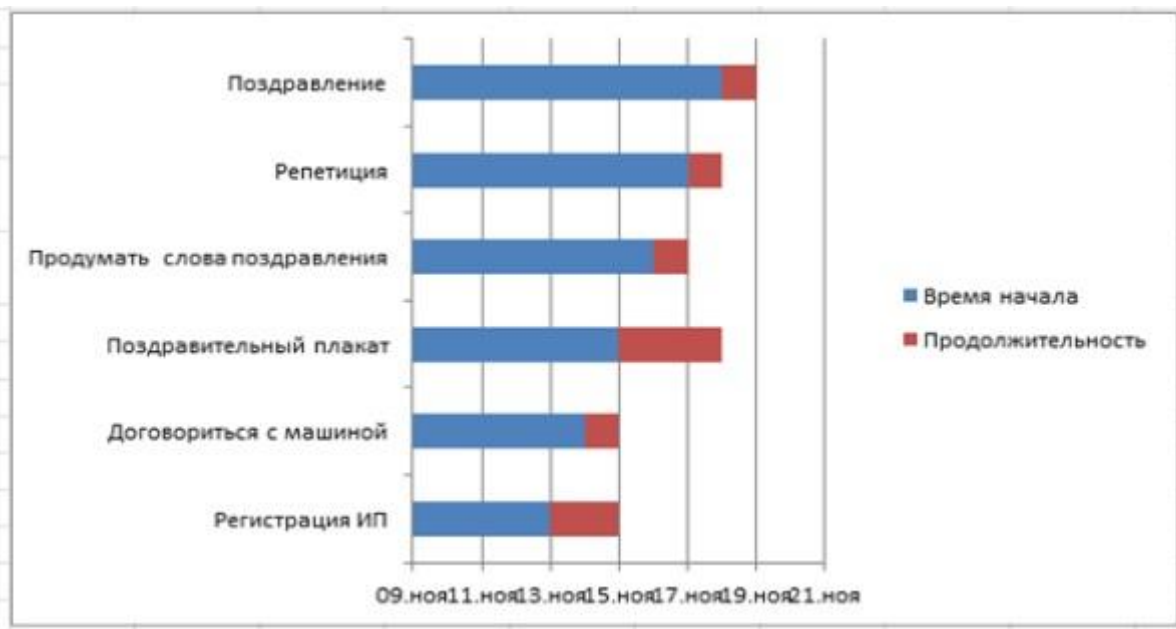


Появляется окно, в котором нужно выделить время начала и нажать кнопку Edit (выделено красным).

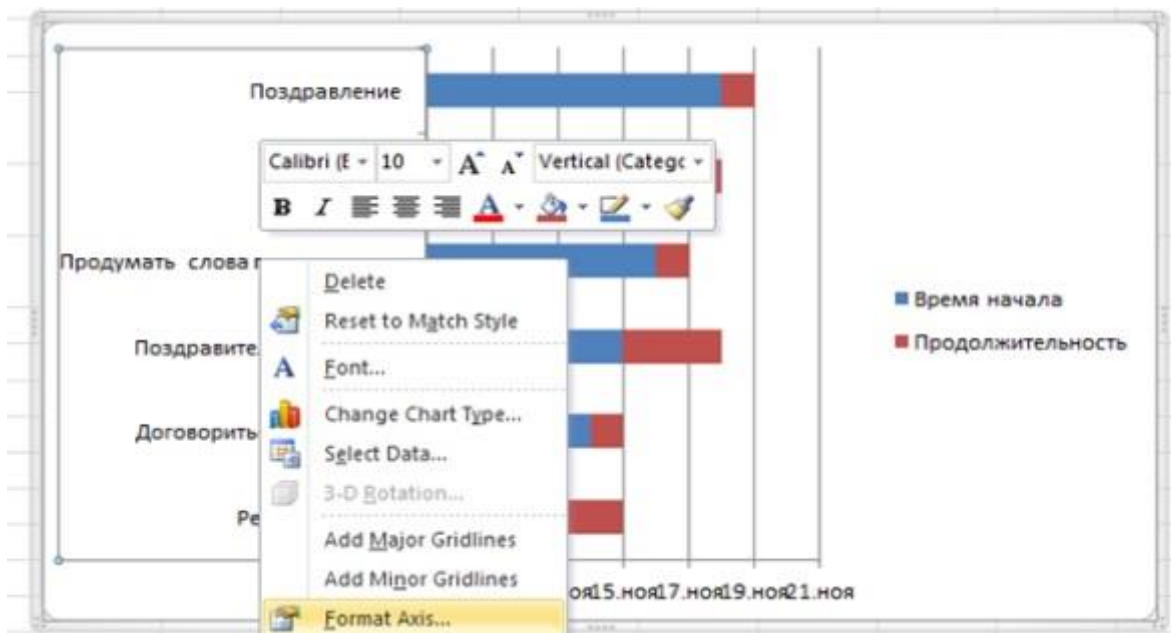


После чего выделить промежуток, откуда планируется брать данные (по аналогии, как это делалось выше) и нажать на кнопку Ok. Как мы видим цифры заменились словами

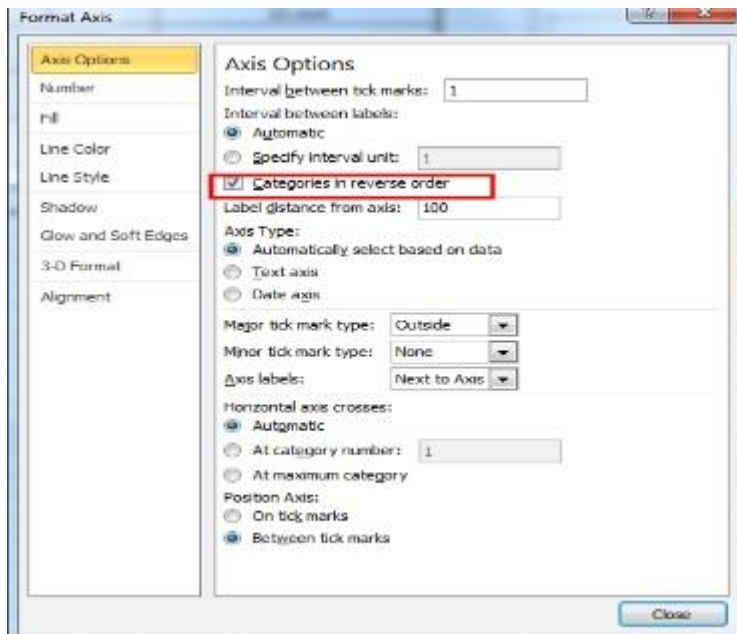




Задачи отображаются в обратном порядке снизу вверх. Для изменения порядка кликаем правой кнопкой мыши по словам. И выбираем опцию Format Axis.



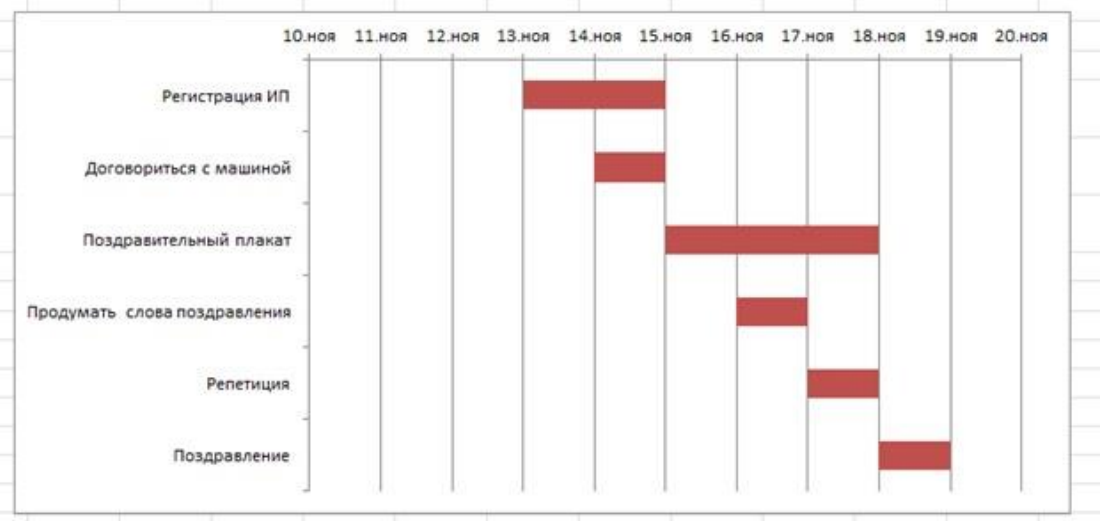
И в появившемся окне ставим галочку в строке выделенной красным.



Следующее, что нужно сделать – это спрятать синие колонки. Для этого выделяем их, после чего кликаем по ним правой клавишей и выбираем опцию **Format Data Series**

В появившемся окне сначала выбираем **Fill->No fill**, потом **Border Color ->No line**. Диаграмма приобрела следующий вид

Для того, чтобы убрать фразы **Время начала** и **Продолжительность** и расширить поле с графиком выбираем и удаляем их. Получается вот такой вот график.



Последнее, что можно сделать – это сдвинуть весь наш график влево, так что отсчет начиналась с 13 ноября.

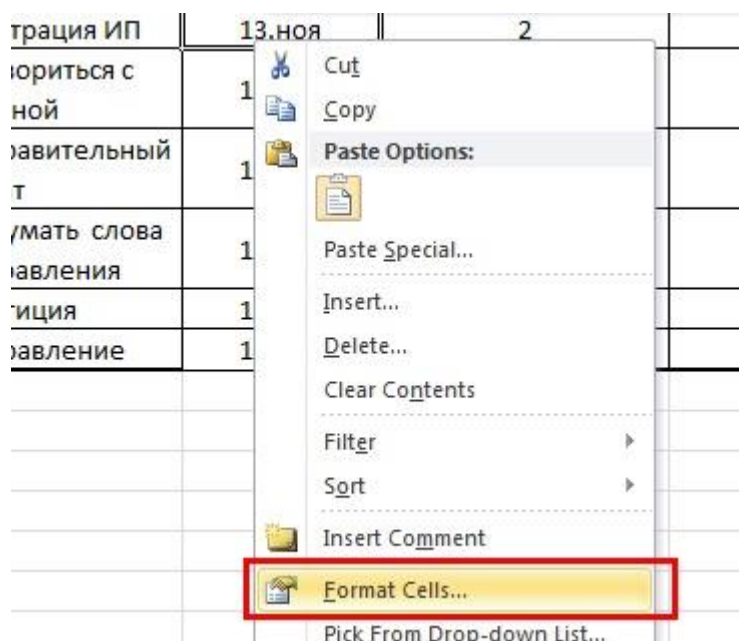
Для это выделяем поле с датами, щелкаем на него правой кнопкой мыши и выбираем опцию **Format Axis**.



В появившемся окне в поле Minimum выбираем Fixed и меняем дату (ее числовое значение) на ту, которая необходима нам.

Как понять, какую дату нужно вписывать в это поле?

Для этого можно щелкнуть правой кнопкой мыши на поле с той датой, которая нам необходима и выбрать опцию Format Cells



Нажать General и в правом углу вы увидите нужную вам дату.

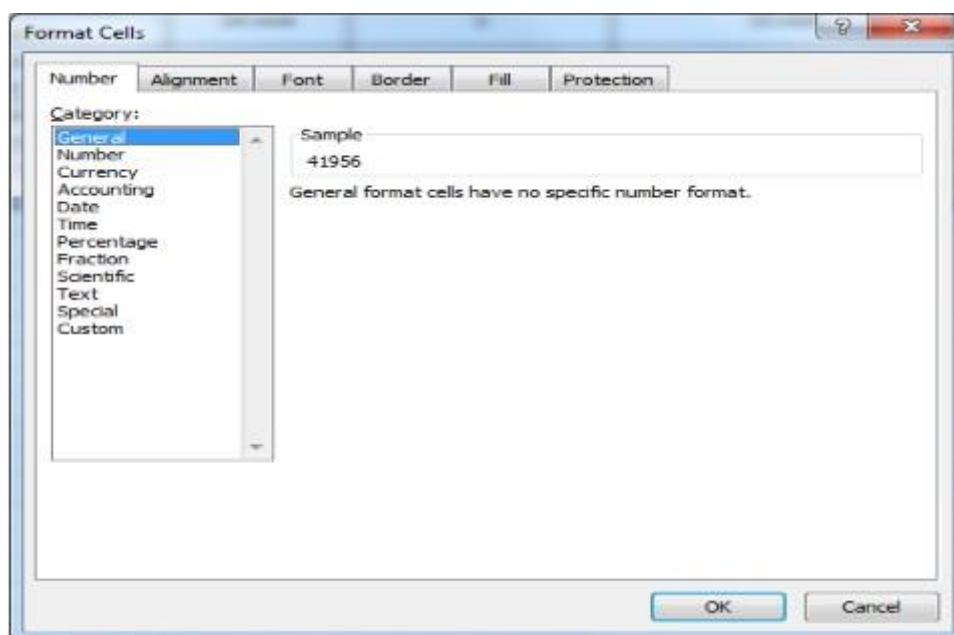


Диаграмма Ганта с помощью Microsoft Excel 2010 построена.

### **3. Задание к лабораторной работе**

Разработать: диаграмму Ганта для реализации проекта разработки информационной системы выбранной предметной области.

#### **Контрольные вопросы**

1. Какие этапы жизненного цикла ИС рассматриваются при составлении графика работ по проекту ИС?
2. С какой целью строится график выполнения работ по проекту?
3. Кто определяет порядок выполнения задач, их продолжительность, время начала и завершения?
4. Каким образом производится оценка срока выполнения проекта в целом?
5. Каким инструментальным средством можно разработать график выполнения проекта ИС?
6. Как создать на диаграмме работу и определить начало и продолжительность работы?
7. Как выбрать вид диаграммы Гантта?
8. Можно ли поменять порядок выполнения работ проекта?
9. От чего зависит срок выполнения работ проекта?
10. Можно ли изменить время реализации проекта?

## Список литературы

1. Лапина, Т. И. Информационные системы. Проектный практикум к выполнению и защите ВКР бакалавра по направлению 09.03.02 Информационные системы, 09.03.03 Прикладная информатика/ Т. И. Лапина//Юго-Западный гос. ун-т, ЗАО «Университетская книга»–Курск, 2016.–99с.
2. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс] : учебное пособие / С. Ю. Золотов. - Томск : Эль Контент, 2013. - 88 с.
3. Абрамов, Г. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / Г. В. Абрамов, И. Медведкова, Л. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с.
4. Стасышин, В. М. Проектирование информационных систем и баз данных [Электронный ресурс] : учебное пособие / В. М. Стасышин. - Новосибирск : НГТУ, 2012. - 100 с.
5. Вендров, А.М. Проектирование программного обеспечения [Текст] : учебник / А.М. Вендров.— М: Финансы и Статистика, 2006. — 352с.

## Практическая работа №5

### Оценка показателей эффективности проекта ИС

#### 1. Цель работы

Получить навыки оценки экономических характеристик выполнения проекта разработки ИС

#### 2. Теоретические сведения

Показатель эффекта от внедрения ИС определяет все позитивные результаты, достигаемые при использовании программного продукта. Прибыль от использования программного продукта за год эксплуатации  $\Pi$ , руб., определяется по формуле

$$\Pi = \text{Э} - \text{З исп.}, \quad (1)$$

где  $\text{Э}$  - стоимостная оценка результатов применения программного продукта в течение года, руб.;

$\text{З исп}$  - стоимостная оценка затрат при использовании программного продукта в течение года, руб.

$$\text{З исп} = 41841,66 \text{ руб.}$$

Приток денежных средств из-за использования программного продукта  $\text{Э}$ , руб., в течение года может составить:

$$\text{Э} = (\text{ЗРУЧ} - \text{ЗАВТ}) + \text{ЭДОП}, \quad (2)$$

где  $\text{ЗРУЧ}$  - затраты на ручную обработку информации, руб.;

$\text{ЗАВТ}$  - затраты на автоматизированную обработку информации, руб.;

$\text{ЭДОП}$  - дополнительный экономический эффект, связанный с уменьшением числа используемых бланков, высвобождением рабочего времени и т. д., руб.

Данный продукт используется офис-менеджером отдела по работе с клиентами фирмы ООО «Софт Центр». Оклад офис-менеджера – 25000руб., премиальный фонд (доп з/п) - 0 от оклада,  $N_{рд} = 22$  дней,  $\Delta t_{рд} = 8$  ч. Тогда, цена одного часа работы офис-менеджера отдела по работе с клиентами цЧ, руб./ч, составит

$$цЧ = (25000 + 25000 \times 0) / 176 = 142,04 \text{ руб./ч.}$$

В ходе исследования было выявлено, что общие затраты времени на ручную обработку информации в месяц составляют 15% общего времени офис-менеджера

$$ОБЩ. Р = 176 * 0.15 = 26,4$$

а общие затраты на автоматизированную обработку информации -  $t_{ОБЩ. А} = 13$  ч.

Годовые затраты (затраты за 12 месяцев) офис-менеджера отдела по работе с клиентами при ручной обработке информации вычислим по формуле

$$ЗРУЧН = t_{ОБЩ. Р} \times 12 \times цЧ = 26,4 \times 12 \times 142,04 = 44985,6 \text{ руб.} \quad (3)$$

Годовые затраты (затраты за 12 месяцев) офис-менеджера при автоматизированной обработке информации вычислим по формуле

$$ЗАВТ = t_{ОБЩ. А} \times 12 \times цЧ = 13 \times 12 \times 142,04 = 22158,24 \text{ руб.} \quad (4)$$

Следовательно, годовой эффект от внедрения программного продукта, даже без учета дополнительного экономического эффекта ( $\text{ЭДОП} = 0$ ), на основании формулы (21) получится равным:

$$\text{Э} = 44985,6 - 22158,24 = 22827,36 \text{ руб.}$$

Эксплуатационные затраты при использовании программного продукта состоят из затрат на электроэнергию, техническое обслуживание, текущий ремонт вычислительно техники и затрат на амортизацию вычислительной техники.

На основании формулы (16), для персонального компьютера офис-менеджера за 12 месяцев затраты на электроэнергию при потребляемой мощности компьютера  $P_B = 0,3$  кВт составят

$$ЗЭ = 0,3 \times 8 \times 12 \times 3,74 = 107,71 \text{ руб.}$$

Балансовая стоимость вычислительной техники  $B_K = 21500,00$  руб. Тогда, на основании формулы (4.18), для персонального компьютера офис-менеджера за 12 месяцев затраты на техническое обслуживание и текущий ремонт составят:

$$ЗП = 21500 \times 0,04 \times (12 \times 8 / 2112) = 39,09 \text{ руб.}$$

Затраты на амортизацию вычислительной техники по формуле (19) составят

$$ЗАО = 21500 \times 0,2 \times (12 \times 8 / 2112) = 195,46 \text{ руб.}$$

Тогда, эксплуатационные затраты при использовании программного продукта составят:

$$ЗИСП = 107,71 + 39,09 + 195,46 = 342,36 \text{ руб.}$$

Прибыль от использования программного продукта за год рассчитаем по формуле (21):

$$П = 22827,36 - 342,36 = 22485 \text{ руб.}$$

Таким образом, имеем следующий денежный поток:

шаг (капиталовложения) - 3 исп = 41841,66 руб.  
руб.;

шаг -22485 руб.;

шаг - 22485 руб.;

шаг - 22485 руб.;

Чистый дисконтированный доход ЧДД, руб., от использования программного продукта определим по формуле:

$$\text{ЧДД} = \sum_{k=1}^n \frac{\Pi}{(1+E)^{k-1}} - K \quad (5)$$

где N - расчетный период, год;

ПК - прибыль от использования программного продукта за k-й год его эксплуатации, руб.;

E - норма дисконта, %;- капиталовложения при внедрении программного продукта, руб.

Следовательно, ЧДД, руб., при N = 3, т. е. за три года использования программного продукта (срок до морального старения рассматриваемой конфигурации) при норме дисконта E = 20% в соответствии с формулой (5) составит:

$$\begin{aligned} \text{ЧДД} &= \frac{22485}{1+0,2} + \frac{22485}{(1+0,2)^2} + \frac{22485}{(1+0,2)^3} - 41841,66 = 18737,5 + 15614,5 + \\ &+ 12997,2 - 41841,66 = 2385,40 \text{ руб.} \end{aligned}$$

Приходим к выводу, что ЧДД – положителен, т. е. проект эффективен.

Рассчитаем срок окупаемости проекта.

Срок окупаемости проекта Ток, год, найдем по формуле

$$T_{ок} = N + \frac{\sum_{j=1}^{N+1} \text{Э}_j - \sum_{j=1}^N \text{Э}_j}{\text{Э}_N} \quad (6)$$

где  $N$  - максимальное количество лет, прошедших с начала эксплуатации программного продукта, в течение которых величина дохода от его использования не превысила величины капиталовложения при внедрении программного продукта;

$\Delta_j$ - величины приведенных (дисконтированных) годовых эффектов за  $j$ -й год, руб., прошедший с начала эксплуатации программного продукта, вычисленные по формуле (5) при подстановке нормы дисконта  $E = 20\%$ .

Величина приведенного (дисконтированного) годового эффекта за первый год расчетного периода по формуле (5) равна:

$$\begin{aligned} \Delta_1 = \frac{22485}{1+0,2} + \frac{22485}{(1+0,2)^2} + \frac{22485}{(1+0,2)^3} = 18737,5 + 15614,5 + \\ + 12997,2 = 47349,2 \text{ руб.} \end{aligned}$$

Величина капиталовложений ( $K = 41841,66$  руб.).

Тогда, в формуле (2) биеем  $N = 0$  и срок окупаемости составит

$$T_{\text{ок}} = 0 + \frac{47349,2}{41841,66} = 1,13 \text{ года.}$$

Внутреннюю норму доходности проекта  $E_{\text{вн}}$ , %, определим по формуле

$$E_{\text{вн}} = E_{\text{вн.МАХ}} + \frac{\text{ЧДД}|_{E_{\text{вн.МАХ}+}}}{\text{ЧДД}|_{E_{\text{вн.МАХ}+}} - \text{ЧДД}|_{E_{\text{вн.МИН}-}}} (E_{\text{вн.МИН}} - E_{\text{вн.МАХ}+}) \quad (6)$$

где  $E_{\text{вн. МАХ} +}$  - максимальное значение внутренней нормы дисконта, %, при которой ЧДД является положительной величиной ( $\text{ЧДД} > 0$ );



ЕВН. MIN - - минимальное значение внутренней нормы дисконта, %, при

которой ЧДД является отрицательной величиной ( $ЧДД < 0$ );

$ЧДД|_{E_{вн. MAX+}}$  - ЧДД, руб., вычисленный по формуле (25) при подстановке нормы дисконта  $E = ЧДД|_{E_{вн. MAX+}}$  ;

$ЧДД|_{E_{вн. MAX-}}$  - ЧДД, руб., вычисленный по формуле (25) при подстановке нормы дисконта  $E = ЧДД|_{E_{вн. MAX-}}$  .

Предполагаем, что ЕВН лежит в диапазоне 174 ... 180%. При норме дисконта ЕВН = 174% получаем ЧДД = 1897,21 руб. Таким образом, при норме дисконта ЕВН = 174% ЧДД - положителен.

При норме дисконта ЕВН = 180% получаем ЧДД = -18014,85 руб. Таким образом, при норме дисконта ЕВН = 180% ЧДД - отрицателен.

Следовательно, по формуле (26) имеем:

$$E_{вн} = 174 + \frac{18940,21}{18940,21 - (-18014,85)} * (180 - 174) = 177,08 \%$$

### 3. Задание к лабораторной работе

Выполнить оценку затрат и полученного эффекта при реализации проекта разработки информационной системы выбранной предметной области.

#### Контрольные вопросы

1. Какие показатели характеризуют экономический результат от внедрения проекта?
2. Как выполнить оценку затрат и полученного эффекта при реализации проекта?

3. Что характеризует внутреннюю норму доходности проекта?
4. Что такое ставка дисконтирования?
5. Что такое приведенный (дисконтированный) годовой эффект?
6. Как рассчитать срок окупаемости проекта?
7. Какой срок приемлем для реализации проекта ИС?
8. Какие проекты можно считать успешными?
9. Что такое эффект от проекта и как он измеряется?
10. Что такое эффективность проекта и как она определяется?

## Практическая работа №6

### Разработка программного интерфейса проекта ИС

#### 1. Цель работы

Получить практические навыки создания программных интерфейсов ИС при помощи Visual Studio и элементов управления Windows - формы.

#### 2. Теоретические сведения

Разрабатываемое приложение предназначено для работы с базой данных сотрудников компании. На рис. 35 представлена структура базы данных.

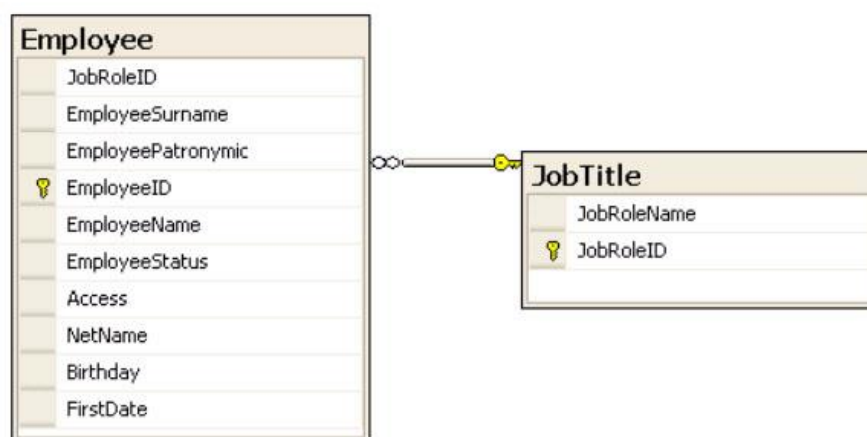


Рисунок 35 – Структура базы данных по сотрудникам компании

Создайте базу данных, включающую две таблицы:

сведения о сотрудниках - *Employee*;  
справочник должностей - *JobTitle*.

Задайте значение атрибутам таблицы *Employee* приведенным в табл. 2

Таблица 2 – Атрибуты таблицы Employee

Имя атрибута	Назначение	Тип
<i>EmployeeID</i>	Суррогатный ключ	smallint
<i>JobRoleID</i>	Внешний ключ	smallint
<i>EmployeeSurname</i>	Фамилия	varchar(50)
<i>EmployeeName</i>	Имя	varchar(20)
<i>EmployeePatronymic</i>	Отчество	varchar(20)
<i>EmployeeStatus</i>	Статус	int
<i>Access</i>	Уровень доступа	varchar(20)
<i>NetName</i>	Сетевое имя	varchar(20)
<i>Birthdate</i>	Дата рождения	Smalldatetime
<i>FirstDate</i>	Дата приема на работу	smalldatetime

Суррогатный ключ *EmployeeID*, как и все остальные суррогатные ключи базы данных, генерируется сервером базы данных автоматически, т.е. для него задано свойство `IDENTITY` для СУБД *MS SQL Server* или `AutoNumber` для *MS Access*. Атрибут *JobRoleID* является внешним ключом, с помощью которого осуществляется связь с табл. *JobTitle*.

Назначение атрибутов таблицы *JobTitle* приведено в табл. 3.

Таблица 3 – Атрибуты таблицы JobTitle

Имя атрибута	Назначение	Тип
<i>JobRoleID</i>	Суррогатный ключ	smallint
<i>JobRoleName</i>	Наименование должности	varchar(50)

В рассматриваемом приложении в качестве СУБД используется *MS SQL Server*.

Создайте соединение проекта с базой данных. Для этого выберите пункт меню *Tools/Connect to Database*.

Появляется окно *AddConnection* (рис.36).

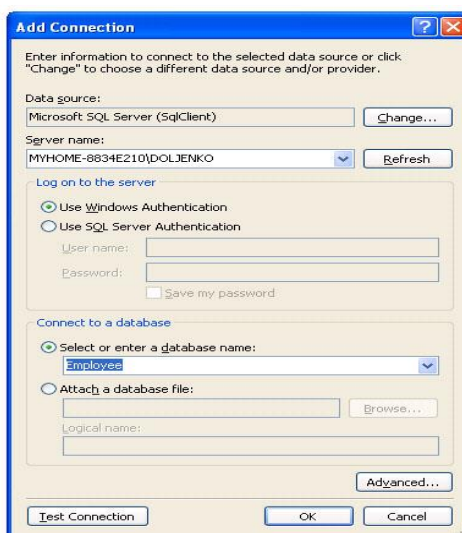


Рисунок 36 – Окно *AddConnection*

В пункте "*Server name*" задайте имя сервера, которое необходимо узнать у преподавателя. В пункте *Select or enter database name* - имя базы данных, которое определит преподаватель.

Для проверки правильности подключения к базе данных нажимаем клавишу "*Test Connection*". При правильном подключении появляется следующее сообщение (рис.37).

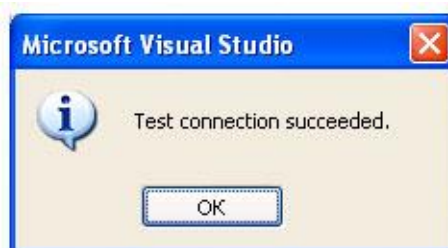


Рисунок 37 – Окно *Microsoft Data Link*

При нормальном соединении с базой данных можно открыть навигатор *Server Explorer* из меню *View/ Server Explorer* или сочетанием клавиш *ALT+CTRL+S* (рис.38).

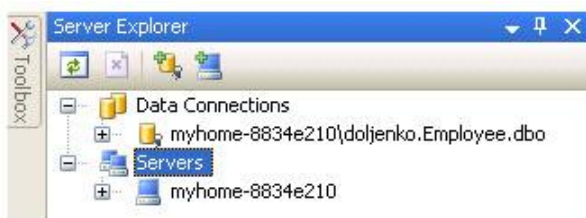


Рисунок 38 – Окно навигатора Server Explorer

Добавьте в проект объект класса DataSet. Для этого выберем пункт меню *Project/Add New Item . .* (рис.39).

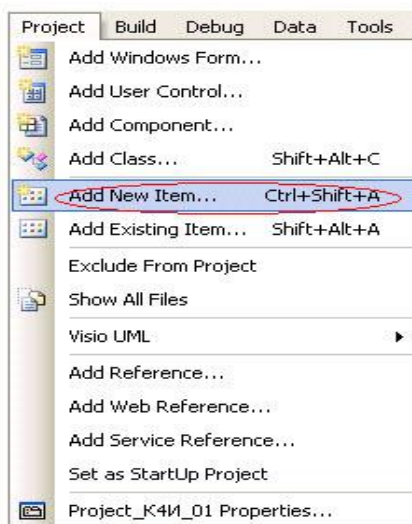


Рисунок 39 – Добавление в проект нового компонента

Выберите шаблон *DataSet* в окне *Add New Item* рис.40 и присвойте ему имя *DataSetEmployee*.

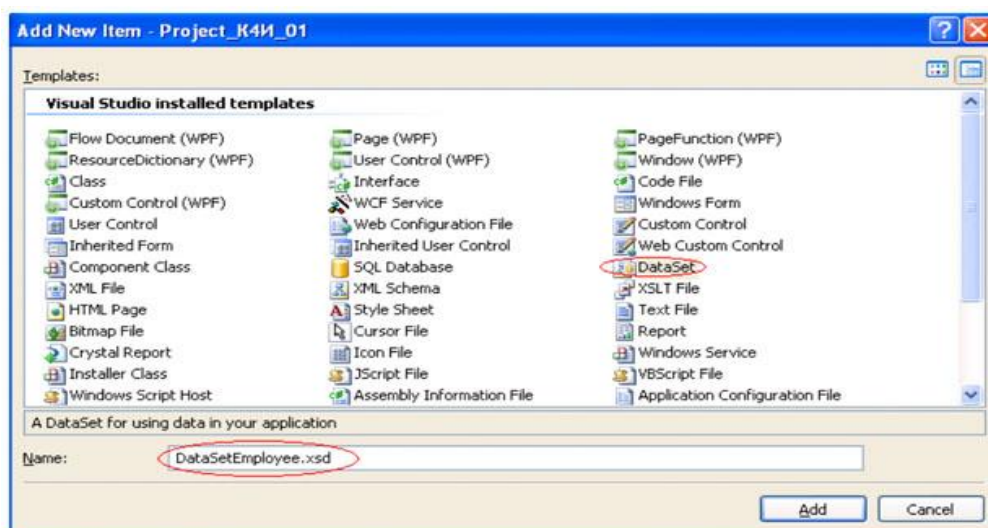


Рисунок 40 – Выбор нового компонента – DataSet

После нажатия кнопки *Add* система генерирует класс *DataSetEmployee*, который добавляется в решение проекта (рис.41).

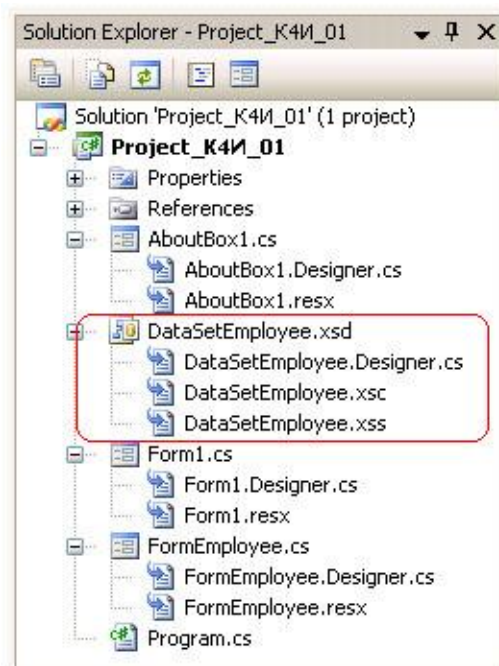


Рисунок 41 – Окно решения проекта с новым компонентом DataSet

Для добавления таблиц *Employee* и *JobTitle* к *DataSet* необходимо перетащить их из окна *Server Explorer* на поле графического дизайнера (рис.42).

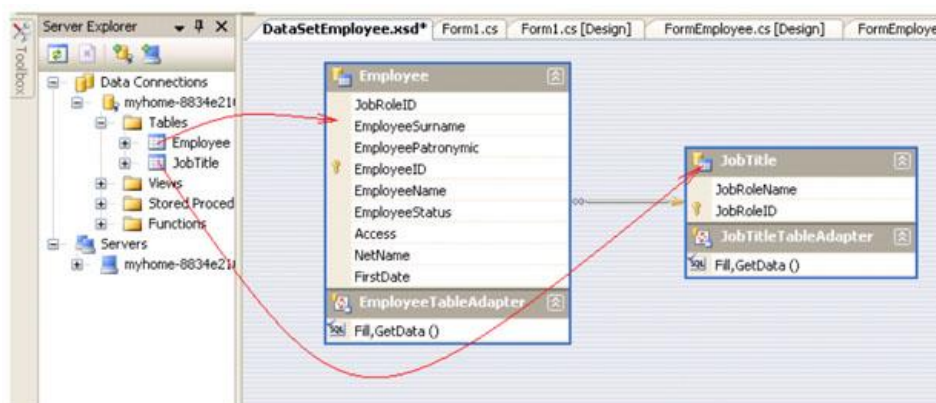


Рисунок 42 – Добавление таблиц к DataSet

В результате будут созданы классы таблиц, адаптеры и методы *Fill* и *GetData*.

При формировании класса `DataSetEmployee` необходимо учесть то, что первичные ключи таблиц *Employee* и *JobTitle* являются суррогатными и автоматически формируются (ключ со свойством автоинкремент) источником данных (например, *MS SQL Server*).

При формировании новых записей в приложении необходимо обеспечить уникальность первичных ключей для таблиц объекта `DataSetEmployee`. Это можно обеспечить, задав для ключевых колонок таблиц *Employee* и *JobTitle* следующие свойства:

```
AutoIncrement = true;  
AutoIncrementSeed = -1;  
AutoIncrementStep = -1;
```

Столбец со свойством `AutoIncrement` равным `true` генерирует последовательность значений, начинающуюся со значения `AutoIncrementSeed` и имеющую шаг `AutoIncrementStep`. Это позволяет генерировать уникальные значения целочисленного столбца первичного ключа. В этом случае при добавлении новой записи в таблицу будет генерироваться новое значение первичного ключа, начиная с -1, -2, -3 и т.д., которое никогда не совпадет с первичным ключом источника данных, т.к. в базе данных генерируются положительные первичные ключи. Свойства `AutoIncrementSeed` и `AutoIncrementStep` устанавливаются равными -1, чтобы гарантировать, что когда набор данных будет синхронизироваться с источником данных, эти значения не будут конфликтовать со значениями первичного ключа в источнике данных. При синхронизации `DataSet` с источником данных, когда добавляют новую строку в таблицу *MS SQL Server 2005* с первичным автоинкрементным ключом, значение, которое этот ключ имел в таблице *DataSet*, заменяется значением, сгенерированным СУБД.

Установите свойства `AutoIncrement`, `AutoIncrementSeed` и `AutoIncrementStep` для колонки первичного ключа `EmployeeID` таблицы *Employee* (рис.43).



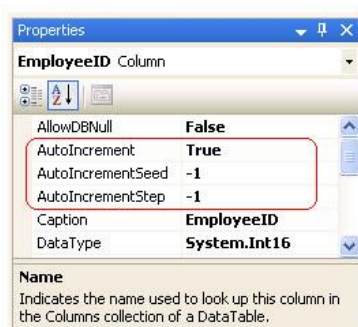


Рисунок 43 – Установка свойств для колонки EmployeeID

Сделайте для колонки `JobTitleID` таблицы *JobTitle*. установки свойств аналогичные `AutoIncrement`, `AutoIncrementSeed` и `AutoIncrementStep`

После создания класса `DataSetEmployee` и адаптера необходимо создать объекты этих классов, добавив следующий код к файлу `FormEmployee.cs`.

```
DataSetEmployee dsEmployee = new DataSetEmployee();
DataSetEmployeeTableAdapters.EmployeeTableAdapter
daEmployee = new Project_K4И_01.DataSetEmployeeTableAdapters.
EmployeeTableAdapter();
DataSetEmployeeTableAdapters.JobTitleTableAdapter daJobTitle =
new Project_K4И_01.DataSetEmployeeTableAdapters.
JobTitleTableAdapter();
```

Создайте метод для заполнения объекта `dsEmployee` из базы данных, после того, как созданы объекты адаптеров данных `daEmployee` и `daJobTitle`, а также объект класса `DataSetEmployee` - `dsEmployee` (в рассматриваемом примере база данных `Employee`, созданная в СУБД *MS SQL Server 2005*). Для заполнения данными `dsEmployee` из базы данных `Employee` создадим метод `EmployeeFill()`:

```
public void EmployeeFill()
{
    daJobTitle.Fill(dsEmployee.JobTitle);
    daEmployee.Fill(dsEmployee.Employee);
    MessageBox.Show("Метод Fill отработал");
}
```

В методе `EmployeeFill()` для объектов класса `DataAdapter` применяется метод `Fill`, который производит заполнение таблиц (*JobTitle* и *Employee*) объекта `dsEmployee` данными из базы данных. Метод `Fill` адаптера данных `DataAdapter` требует указания в качестве параметров задания соответствующей таблицы *DataSet*, то есть `dsEmployee.JobTitle` и `dsEmployee.Employee`.

Метод `MessageBox.Show` введен в метод `EmployeeFill` для первоначального тестирования, после которого его нужно убрать.

Вызов метода `EmployeeFill` необходимо добавить в обработчик события `Load` для формы `FormEmployee`, возникающего при нажатии на пункт меню "Сотрудник".

### **Пример выполнения задания:**

1. Свяжите элемент контроля `listBoxEmployee`, в котором должен отображаться список фамилий сотрудников, со столбцом `EmployeeSurname` таблицы `Employee`. Это можно сделать, добавив следующие строки кода в метод загрузки формы `FormEmployee_Load`.

```
this.listBoxEmployee.DataSource = this.dsEmployee1;  
this.listBoxEmployee.DisplayMember = "Employee.EmployeeSurname";
```

В разрабатываемом приложении на *Windows*-форме *FormEmployee* имеются четыре текстовых поля: `textBoxSurname`, `textBoxName`, `textBoxPatronymic` и `textBoxNetName`. Эти текстовые поля предназначены для отображения информации из одной записи таблицы `Employee` набора данных `dsEmployee`. Для того чтобы содержимое текстовых полей автоматически обновлялось при смене записи, их необходимо связать с соответствующими колонками набора данных `dsEmployee`. Связывание можно осуществить, используя свойство `DataBindings` элемента управления `TextBox`.

2. Создайте связь для элемента управления `textBoxSurname` с источником данных, добавив следующую строку кода в метод загрузки формы `FormEmployee_Load`.  
`textBoxSurname.DataBindings.Add("Text", dsEmployee, "Employee.EmployeeSurname");`
3. Протестируйте добавленный в программу код.
4. Аналогично свяжите текстовые поля `textBoxName`, `textBoxPatronymic` и `textBoxNetName` с источником данных.

В таблице `Employee` значения для атрибута `Access` (доступ) задается в виде символьной строки, значение которой выбирается из списка элемента управления `comboBoxAccess`.

5. Задайте коллекцию выпадающего списка элемента управления `comboBoxAccess` можно задать следующей строкой кода:  
`this.comboBoxAccess.Items.AddRange(new object[] { "не задан", "администратор", "начальник смены", "старший оператор", "оператор", "аналитик" });`
6. Для заданной записи источника данных (таблица `Employee`) значение столбца `Access` отобразите в элементе контроля `comboBoxAccess`. Сделайте это аналогично тому, как это делалось для элементов управления `TextBox`, задавая свойство `DataBindings`
7. Сформируйте коллекцию выпадающего списка: не задан, активен, выходной, в отпуске, болеет, не работает, помечен как удаленный. Для элемента управления `comboBoxStatus`

В таблице `Employee` значения для атрибута `EmployeeStatus` (статус) задается в виде целого числа (0, 1, 2, 3, 4, 6), однако статус сотрудника должен отображаться в элементе управления `comboBoxStatus` в виде строковых значений в соответствии со значениями его коллекции. В программе необходимо реализовать отображение целочисленных данных из `DataSet` в текстовые значения в элементе контроля `comboBoxStatus`. Для этого необходимо отслеживать изменение позиции в табл. источника данных `dsEmployee` и в соответствии со значением столбца `EmployeeStatus` активизировать требуемый элемент (`Item`) списка `comboBoxStatus`.

8. Объявите объект `bmEmployee` класса `BindingManagerBase` в форме *ormEmployee*:

```
BindingManagerBase bmEmployee;
```

9. Создайте объект `bmEmployee` в конструкторе класса `FormEmployee`, применяя индекатор контента `BindingContext` включив в него связывание с таблицей `Employee`, и добавьте делегат для события, которое формируется при изменении позиции в данной таблице:

```
public FormEmployee( )
```

```
{ InitializeComponent();
```

```
  bmEmployee = this.BindingContext[dsEmployee1, "Employee"];
```

```
  // Добавляем делегата PositionChanged для события - изменение
```

```
  //позиции в табл. Employee DataSet dsEmployee
```

```
  bmEmployee.PositionChanged = new EventHandler(BindingManager-  
Base_PositionChanged); }
```

10. Создайте обработчик для сформированного события, который на основе выбранной строки (`pos`) таблицы `Employee` будет задавать свойству `Text` списка `comboBoxStatus` значение из коллекции `Items` по индексу (`sel`), полученному из столбца `EmployeeStatus` `Employee`.

```
private void BindingManagerBase_PositionChanged(object sender,  
EventArgs e)
```

```
{
```

```
  int pos = ((BindingManagerBase)sender).Position;
```

```
  int sel = (int)dsEmployee.Employee[pos].EmployeeStatus;
```

```
  this.comboBoxStatus.Text = this.comboBoxSta-
```

```
  tus.Items[sel].ToString();
```

```
}
```

11. **Протестируйте добавленный в программу код.**

12. Для задания в элементе контроля `comboBoxJobRole` должности сотрудника необходимо получить данные из родительской таблицы `JobTitle`, с которой таблица `Employee` связана внешним ключом `JobRoleID`. Фактически необходимо осуществить вывод данных из справочника (таблица `JobTitle`) по данным в основной табл. `Employee`.

Свойство `DataBindings` объекта `ComboBox` предоставляет доступ к коллекции `ControlBindingsCollection`. Метод `Add` этой коллекции добавляет в неё привязку.

Перегруженный вариант метода `Add` принимает три аргумента:

`PropertyName` - имя свойства элемента управления, к которому осуществляется привязка;

`DataSource` - имя привязываемого источника данных;

`DataMember` - имя свойства привязываемого источника данных.

13. Свяжите элемент контроля `comboBoxJobRole` с набором данных `JobTitle` (родительская таблица - справочник), в соответствии с кодом, приведенным ниже.

```
comboBoxJobRole.DataSource = this.dsEmployee.JobTitle;
```

```
comboBoxJobRole.DisplayMember = "JobRoleName";
```

```
comboBoxJobRole.ValueMember = "JobRoleID";
```

14. Свяжите `comboBoxJobRole` с полем `JobRoleID` основной таблицы `Employee` (дочерняя таблица), в соответствии со следующим кодом

```
comboBoxJobRole.DataBindings.Add("SelectedValue", dsEmployee, "Employee.JobRoleID");
```

После компиляции и запуска приложения экранная форма будет иметь вид, приведенный на рис.44.

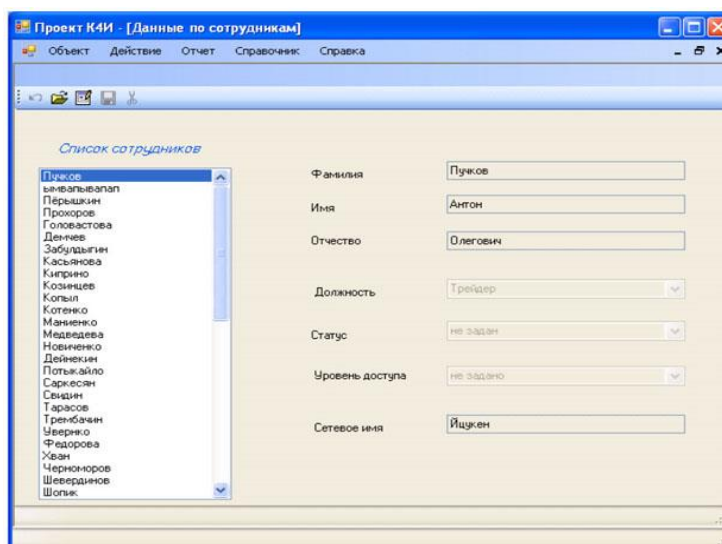


Рисунок 44 – Экранная форма в режиме просмотра (только для чтения)

### 3. Задание к лабораторной работе

1. Опишите процесс создание класса DataSetEmployee, объектов dsEmployee, daJobTitle и daEmployee, метода Fill.

2. Опишите создания базы данных, включающей две таблицы и соединение проекта с базой данных, формирование новых записей в приложении.

3. Создайте метод для заполнения объекта dsEmployee из базы данных.

4. Опишите процесс связывания элемента контроля listBoxEmployee, в котором должен отображаться список фамилий сотрудников, со столбцом EmployeeSurname таблицы Employee.

5. Выводы по проделанной работе

#### Контрольные вопросы

1. Как подготовить таблицы базы данных?
2. Как использовать базы данных в приложении?
3. Как создать соединение проекта с базой данных?
4. Как использовать класс DataSetEmployee?
5. Что такое объекты dsEmployee, daJobTitle и daEmployee?
6. Какие функции выполняет метод Fill?
7. Как создать метод для заполнения объекта dsEmployee из базы данных?
8. Какой порядок процесса связывания элемента контроля listBoxEmployee, в котором должен отображаться список фамилий сотрудников, со столбцом EmployeeSurname таблицы Employee?
9. Что представляет собой функция Data.Set?
10. Для выполнения, каких задач предназначены все типы ADO.NET?

## Список рекомендуемых источников

1. Проектирование информационных систем. Проектный практикум : учебное пособие / А. В. Платёнкин, И. П. Рак, А. В. Терехов, В. Н. Чернышов ; Тамбовский государственный технический университет. - Тамбов : Тамбовский государственный технический университет (ТГТУ), 2015. - 81 с. : ил., схем. - URL: <http://biblioclub.ru/index.php?page=book&id=444966> (дата обращения 27.02.2020) . - Режим доступа: по подписке. - Библиогр. в кн. - ISBN 978-5-8265-1409-2. - Текст : электронный.
2. Антонов, В. Ф. Методы и средства проектирования информационных систем : учебное пособие / В. Ф. Антонов, А. А. Москвитин ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2016. – 342 с. - URL: <http://biblioclub.ru/index.php?page=book&id=458663>. - Режим доступа: по подписке. - Текст : электронный.
3. Влацкая, И. В. Проектирование и реализация прикладного программного обеспечения [Электронный ресурс] : учебное пособие / И. В. Влацкая ; Н. А. Заельская ; Н. С. Надточий. - Оренбург : ОГУ, 2015. - 119 с. – Режим доступа: [biblioclub.ru](http://biblioclub.ru)
4. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс] : учебное пособие / С. Ю. Золотов. - Томск : Эль Контент, 2013. - 88 с. // Режим доступа – <http://biblioclub.ru/index.php?page=book&id=208706>
5. Абрамов, Г. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / Г. В. Абрамов, И. Медведкова, Л. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с. //Режим доступа –<http://biblioclub.ru/index.php?page=book&id=141626>
6. Стасышин, В. М. Проектирование информационных систем и баз данных [Электронный ресурс] : учебное пособие / В. М. Стасышин. - Новосибирск : НГТУ, 2012. - 100 с. // Режим доступа – <http://biblioclub.ru/index.php?page=book&id=228774>

8. Романов, В. П. Проектирование экономических информационных систем. Методология и современные технологии [Текст] : учебное пособие / В. П. Романов, Н. З. Емельянова, Т. Л. Партыка ; Российская экономическая академия им. Г. В. Плеханова. - М. : Экзамен, 2005. - 256 с. - (Учебник Плехановской академии). - ISBN 5-472-00742-9.

9. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем [Текст] : учебник для студ. вуз. / А. М. Вендров. - М.: Финансы и статистика, 2000. - 352 с. : ил. - ISBN 5-279-02144-X.

10. Вендров, А. М. Практикум по проектированию программного обеспечения экономических информационных систем [Текст] : учебное пособие / А. М. Вендров. - М. : Финансы и статистика, 2002. - 192 с. - ISBN 5-279-02440-6.

11. Торрес, Р. Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса [Текст] / Р. Дж. Торрес. - М. : Вильямс, 2002. - 400 с.

12. Лапина, Т. И. Информационные системы. Проектный практикум к выполнению и защите ВКР бакалавра по направлению 09.03.02 Информационные системы, 09.03.03 Прикладная информатика/ Т. И. Лапина//Юго-Западный гос. ун-т, ЗАО «Университетская книга»—Курск, 2016.—99с.