

Государственное бюджетное профессиональное образовательное учреждение
«Кунгурский автотранспортный колледж»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ПО ВЫПОЛНЕНИЮ ВНЕАУДИТОРНЫХ
САМОСТОЯТЕЛЬНЫХ РАБОТ**

по междисциплинарному курсу

**ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

по специальности

09.02.07 Информационные системы и программирование

СОДЕРЖАНИЕ

Пояснительная записка	3
Порядок выполнения практической работы	4
Перечень практических работ	5
Самостоятельная работа №1 «Организационная схема и диаграмма окружения функции»	6
Самостоятельная работа №2 «Дерево функций ARIS»	9
Самостоятельная работа №3 «Событийная цепочка процесса (цепочка событий). Построение средствами MS Visio »	12
Список литературы.....	15

Пояснительная записка

Методические указания по выполнению внеаудиторных самостоятельных работ по МДК 01.01 Разработка программных модулей разработаны в соответствии с рабочей программой профессионального модуля и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении профессионального модуля, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

МДК 01.01 Разработка программных модулей относится к профессиональному циклу, изучается на 2 курсе и при его изучении отводится значительное место выполнению внеаудиторных самостоятельных работ.

Освоение содержания МДК 01.01 Разработка программных модулей во время выполнения практических работ обеспечивает достижение обучающимися следующих **результатов:**

Код	Наименование общих компетенций
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием.

В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	В разработке кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений
Уметь	осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства
Знать	основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов

В соответствии с рабочей программой ПМ.01. Разработка модулей программного обеспечения для компьютерных систем внеаудиторные самостоятельные работ по МДК 01.01 Разработка программных модулей разбиты на два семестра – пятый и шестой семестры. Целесообразность данной группировки обусловлена необходимостью обобщения и систематизации знаний перед защитой курсовой работы и дачей экзамена.

Рабочая программа профессионального модуля предусматривает проведение практических работ в объеме 12 часов.

Порядок выполнения практической работы

- записать название работы, ее цель в тетрадь;
- выполнить основные задания в соответствии с ходом работы;
- выполнить индивидуальные задания.

Рекомендации по оформлению практической работы

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе

Результаты выполнения практических работ необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

Критерии оценки практической работы

- наличие цели выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие цели выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие цели выполняемой работы, выполнение всех основных и индивидуальных заданий (отлично).

Перечень внеаудиторных самостоятельных работ

№	Наименование разделов и тем профессионального модуля (ПМ)	Виды ВСР	Объем часов
1	Курсовое проектирование	Самостоятельная работа №1 «. Организационная схема и диаграмма окружения функции»	4
2		Самостоятельная работа №2 «Дерево функций ARIS»	4
3		Самостоятельная работа №3 «Событийная цепочка процесса (цепочка событий). Построение средствами MS Visio»	4
Итого			12

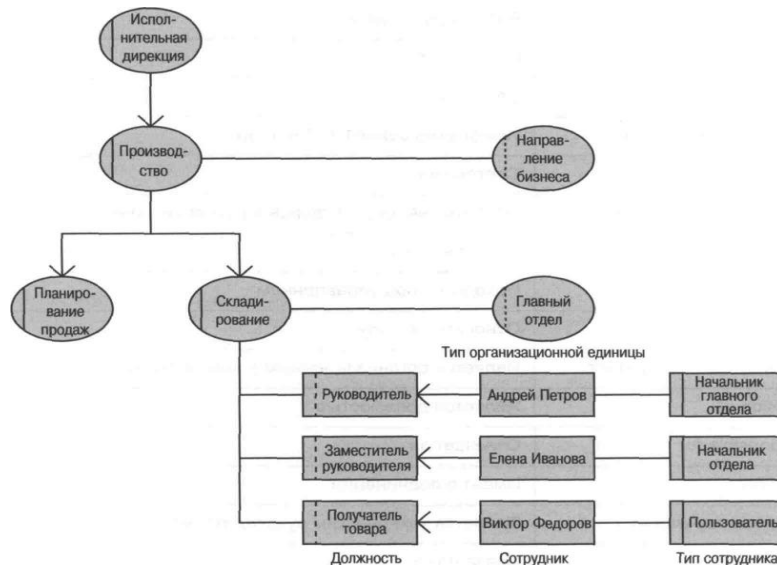
Самостоятельная работа № 1. Организационная схема и диаграмма окружения функции средствами MS Visio

Цель работы: Практическое знакомство с приемами и методами использования диаграмм в нотации ARIS. Организационная схема, дерево функций, диаграмма окружения функции.
Стимул. Вы работаете на предприятии в отделе проектирования программного обеспечения. Перед вами поставлена задача выполнить всестороннее обследование предметной области, по которой вам выдано индивидуальное задание. Результаты обследования необходимо представить в виде моделей в нотации ARIS и UML. По окончании работы требуется представить презентацию со всеми моделями и описание предметной области. Следующие 8 практических занятий посвящены отработке навыков по данной тематике.

Ход работы

Часть 1. Организационная схема ARIS

Задание 1. Рассмотреть организационную структуру предприятия, проанализировать уровни подчинения.



Представить данную схему в нотации организационных схем ARIS, используя соответствующие обозначения.

Задание 2. Задание для самостоятельной работы студентов. Записать в тетрадь формулировку своей предметной области. По образцу орг. структуры предприятия построить организационную схему для своей предметной области.

Порядок работы в MS Visio.

1. Запустите редактор MS Visio (Пуск – Программы – Microsoft Office - Microsoft Office Visio 2007).
2. В появившемся окне выбираем шаблон Бизнес – Организационная диаграмма (рис.1) - Создать.



Рисунок 1 – Выбор шаблона Организационная диаграмма

3. Все элементы диаграммы находятся в левом меню программы (рис. 2). Для создания элемента достаточно потянуть его из меню на страницу.

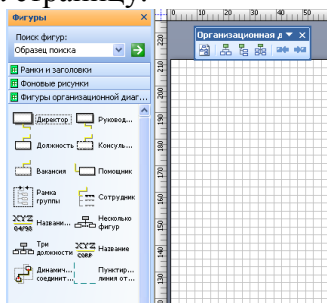


Рисунок 2 –Меню Фигуры

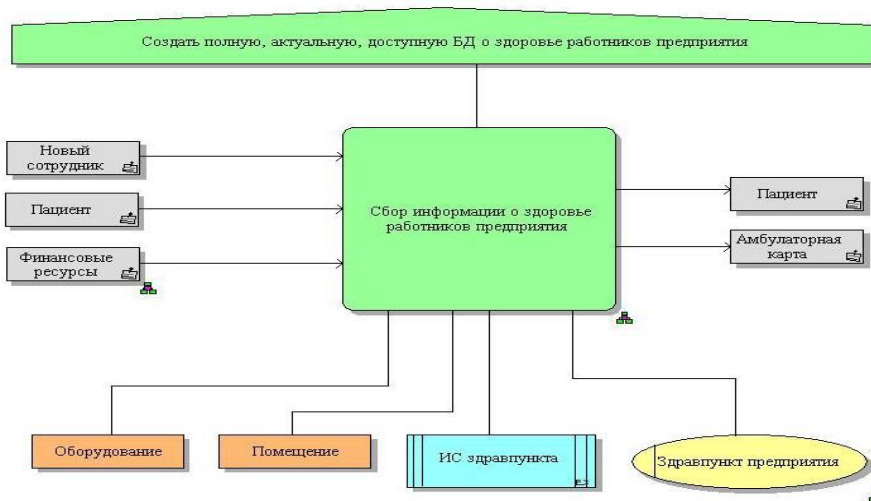
4. При выборе фигуры, внутри появится слова «Имя» и «Должность», которые необходимо поменять. Для изменения текста щелкните два раза по элементу.
5. Для создания орг. структуры выбирайте соответствующие элементы из левого меню. Чтобы создать связь нужно потянуть подчиненный элемент на элемент-родитель (элемент «Андрей Петров» тянем на элемент «Начальник главного отдела»).

Часть 2. Диаграмма окружения функции

Задание 1. Диаграмма окружения функции предназначена для того, чтобы описать все объекты, которые окружают функцию:

- 1) исполнителей,
- 2) входные и выходные потоки информации,
- 3) документы, материалы, продукты/услуги,
- 4) а так же используемое оборудование.

Рассмотрите данную функциональную схему и определите уровни детализации функций:



Задание 2. Задание для самостоятельной работы студентов. Создать диаграмму окружения для главной функции своей предметной области.

Порядок работы в MS Visio.

1. Запустите редактор MS Visio (Пуск – Программы – Microsoft Office - Microsoft Office Visio 2007).
2. В появившемся окне выбираем шаблон Бизнес – Схема ERC (рис.1) - Создать.

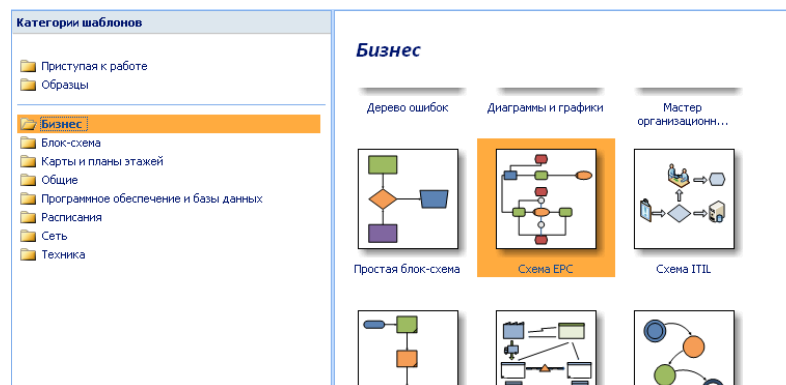


Рисунок 1 – Выбор шаблона Схема EPC

3. Все элементы диаграммы находятся в левом меню программы (рис. 2). Для создания элемента достаточно потянуть его из меню на страницу.

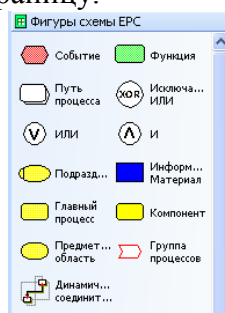


Рисунок 2 –Меню Фигуры

Контрольные вопросы:

1. Укажите главную функцию.
2. Укажите, сколько уровней имеет иерархическая структура для вашей предметной области.
3. Укажите обозначения для структурных единиц и для исполнителей
4. Сколько уровней подчинения в орг. структуре вашей предметной области.

Оформите отчет. Сделайте выводы по работе.

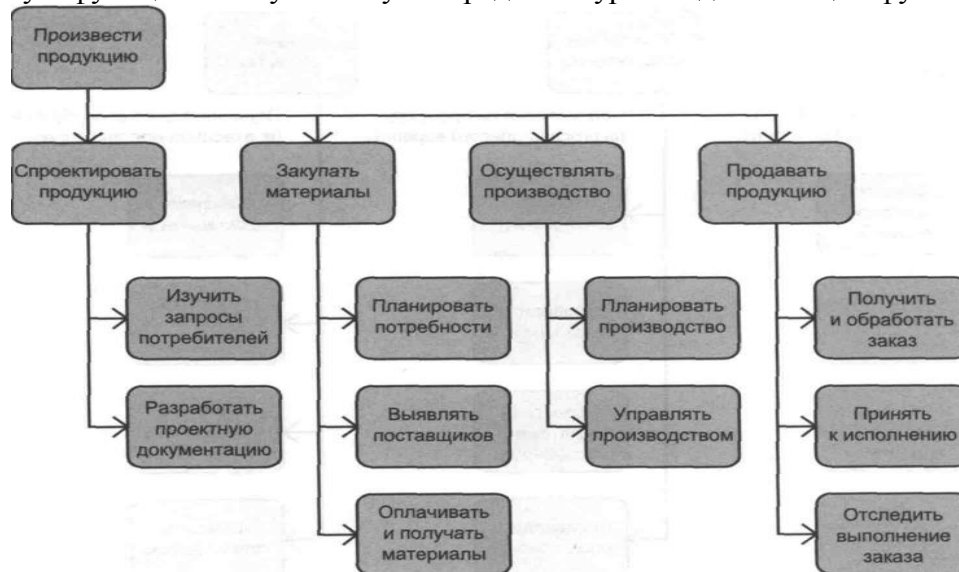
Самостоятельная работа №2. Дерево функций ARIS

Цель работы: ознакомиться с приемами и методами построения функциональных схем в виде дерева функций ARIS.

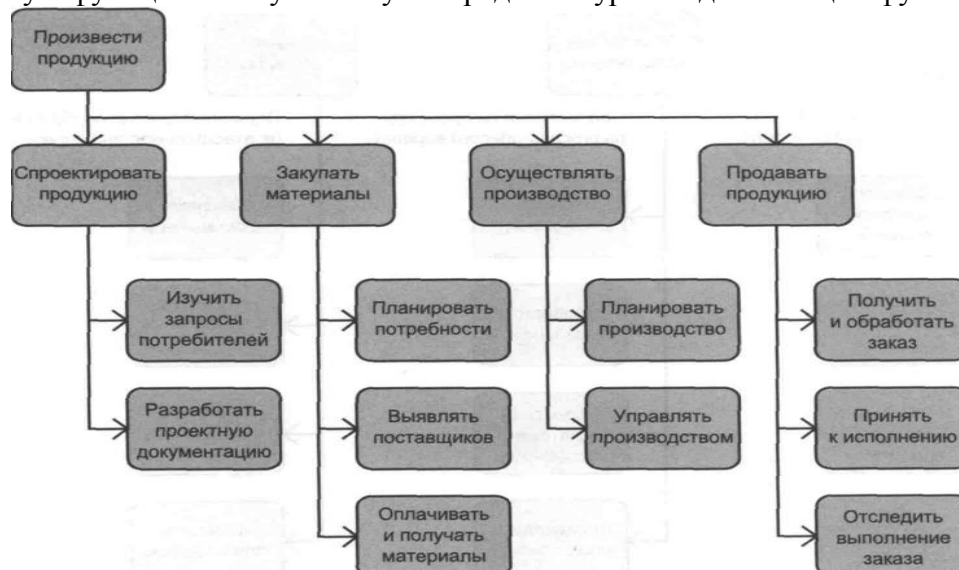
Ход работы

Часть 2. Дерево функций ARIS

Задание 1. Функция — один логический этап в рамках процесса. В ARIS используется диаграмма «Дерево функций», где функции могут быть описаны с различными уровнями детализации. На диаграммах функция обозначается прямоугольником со скругленными краями. Рассмотрите данную функциональную схему и определите уровни детализации функций:

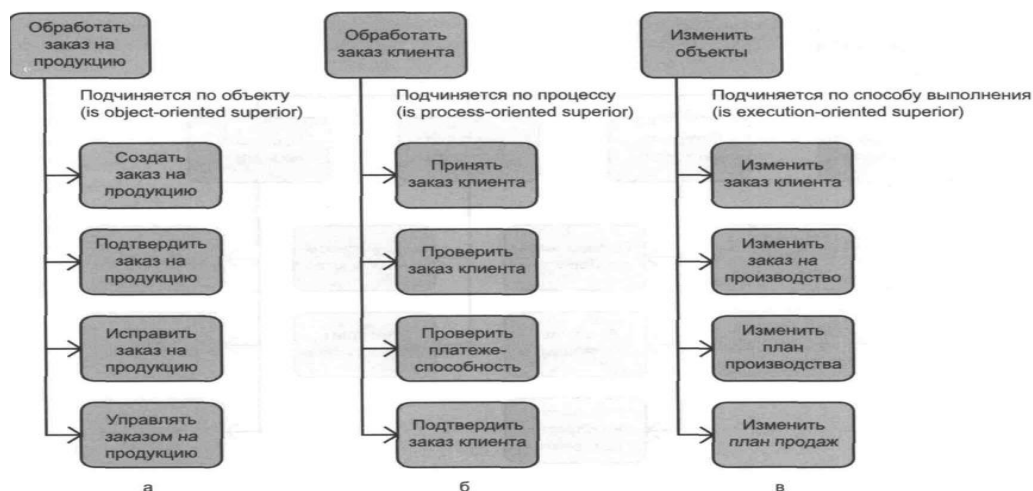


1. **Функция** — один логический этап в рамках процесса. В ARIS используется диаграмма «Дерево функций», где функции могут быть описаны с различными уровнями детализации. На диаграммах функция обозначается прямоугольником со скругленными краями. Рассмотрите данную функциональную схему и определите уровни детализации функций:



2. **Функции объединяются в функциональное дерево по различным критериям:**
- **объектно-ориентированный** — обработка одного и того же объекта;
 - **процессно-ориентированный** — принадлежность одному и тому же процессу;
 - **операционно-ориентированный** — выполнение одинаковых операций.

Например, на рисунке дерево функций: объектно-ориентированное (а); процессно-ориентированное (б); операционно-ориентированное (в)



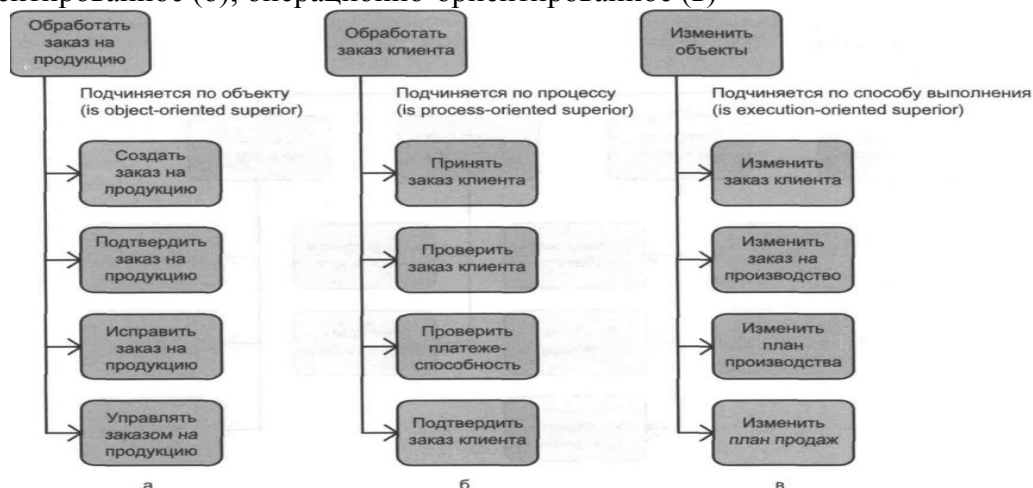
3. **Задание для самостоятельной работы студентов.** Записать в тетрадь формулировку своей предметной области. Выбрать критерий объединения функций в деревья и по образцу приведенной функциональной модели построить дерево функций для своей предметной области. Зачет дифференцированный. Критерии оценки: 1,2 или 3 построенных дерева функций по различным критериям, корректность, полнота, своевременность и аккуратность оформления, ответы на контрольные вопросы, вывод по работе.

Задание 2.

Функции объединяются в функциональное дерево по различным критериям:

- **объектно-ориентированный** — обработка одного и того же объекта;
- **процессно-ориентированный** — принадлежность одному и тому же процессу;
- **операционно-ориентированный** — выполнение одинаковых операций.

Например, на рисунке дерево функций: объектно-ориентированное (а); процессно-ориентированное (б); операционно-ориентированное (в)



Задание 3. Задание для самостоятельной работы студентов. Записать формулировку своей предметной области. Выбрать критерий объединения функций в деревья и по образцу приведенной функциональной модели построить дерево функций для своей предметной области. Зачет дифференцированный. Критерии оценки: 1,2 или 3 построенных дерева функций по различным критериям, корректность, полнота, своевременность и аккуратность оформления, ответы на контрольные вопросы, вывод по работе.

Порядок работы в MS Visio.

1. Запустите редактор MS Visio (Пуск – Программы – Microsoft Office - Microsoft Office Visio 2007).
2. Так как в редакторе MS Visio не существует шаблона дерева функций, то в появившемся окне выбираем шаблон Программное обеспечение и базы данных – Схема потомков данных (рис.1) - Создать.

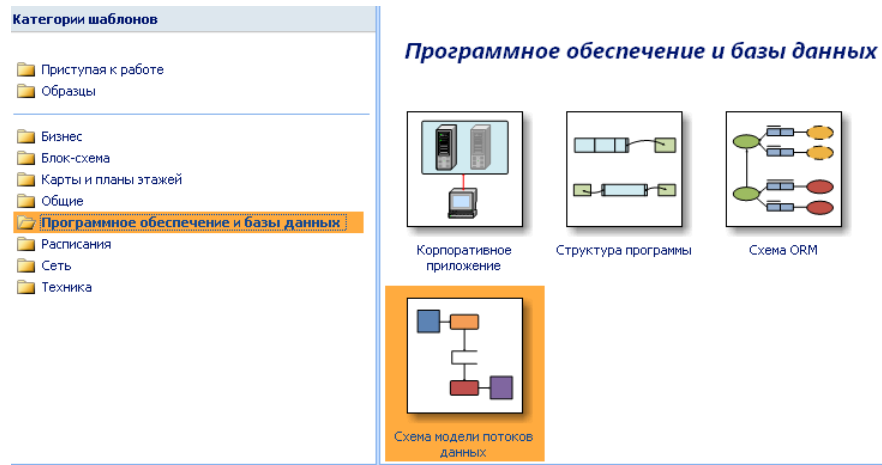


Рисунок 2 – Выбор шаблона Схема модели потоков данных

3. Все элементы диаграммы находятся в левом меню программы (рис. 2). Для создания элемента достаточно потянуть его из меню на страницу. Нас будут интересовать только два элемента – это «Процесс» и «Поток данных».

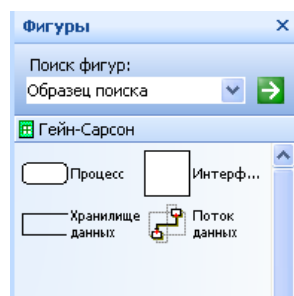


Рисунок 3 –Меню Фигуры

4. Для добавления текста в фигуру щелкните по ней два раза.
 5. Для создания связей между элементами выбираем элемент «Поток данных» и наводим его сначала на элемент-потомок, а затем перемещаем конец стрелки на подчиненный элемент (при этом элементы будут выделяться красным цветом при наведении).

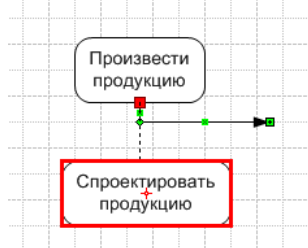


Рисунок 4 – создание связи между элементами

4. **Контрольные вопросы:**

- 1) Укажите главную и подчиненные функции.
 - 2) Укажите, сколько уровней имеет иерархическая структура функции вашей предметной области.
- Сделайте выводы по работе.**

Самостоятельная работа №3. Событийная цепочка процесса (цепочка событий). Построение средствами MS Visio

Цель работы: практически освоить приемы и методы построения событийных цепочек в нотации ARIS средствами MS Visio.

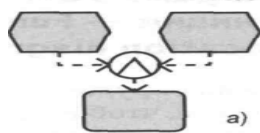
Необходимые сведения. Правила расположения графических элементов на диаграмме :

- 1) последовательность событий и функций располагаются сверху вниз;
- 2) элементы, обозначающие исполнителей функций (сотрудников или подразделения) располагаются справа от функций;
- 3) документы располагаются слева от функций.

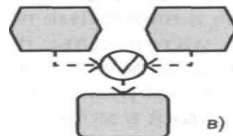
Одно событие может инициировать выполнение нескольких функций, и наоборот, функция может быть результатом наступления нескольких событий.

События инициируют функции

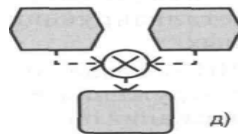
Оператор *AND* (И, обозначается \wedge) для переключающих событий: Выполнение функции после того, как произойдут все события.



Оператор *OR* (ИЛИ, обозначается \vee) для переключающих событий: Эта функция выполняется, если произойдет по крайней мере одно событие.

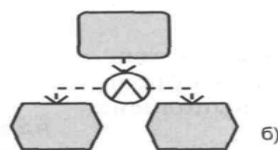


Оператор *XOR* (исключающий ИЛИ, обозначается \otimes) для переключающих событий: Функция начинает выполняться после того, как произойдет одно (и только одно) событие, оба события не могут произойти одновременно.

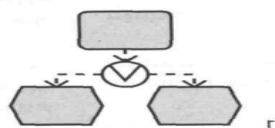


Для соединения сгенерированных событий:

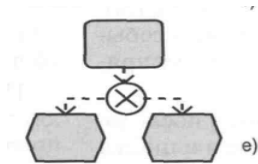
Оператор *AND* для сгенерированных событий: В результате выполнения функции происходят все события.



Оператор *OR* для сгенерированных событий: В результате выполнения функции происходит по крайней мере одно событие или несколько.

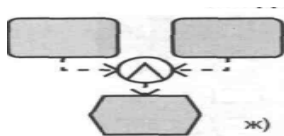


Оператор *XOR* для сгенерированных событий: В результате выполнения функции происходит максимум одно событие.

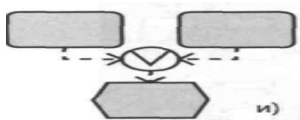


Соединение функций со сгенерированными событиями.

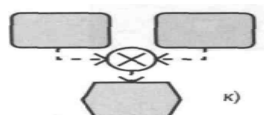
Оператор AND для связи функций и сгенерированных событий. События происходят после того, как все функции выполнены.



Оператор OR. Событие произойдет после того, как будет выполнена по крайней мере одна функция.



Оператор XOR. Событие произойдет после того, как будет выполнена одна (и только одна) функция.



Операторы OR и XOR не могут использоваться для соединения переключающих событий и функций, так как события не могут принимать решения.

Ход работы

1. Постройте примеры на каждый вид соединений от а) до к). Желательно привести эти примеры из своей предметной области. Если это затруднительно, то из любой другой.
2. Задание для самостоятельной работы студентов. Записать формулировку своей предметной области. Построить цепочку событий для своей предметной области.

Порядок работы в MS Visio.

4. Запустите редактор MS Visio (Пуск – Программы – Microsoft Office - Microsoft Office Visio 2007).
5. В появившемся окне выбираем шаблон Бизнес – Схема ERC (рис.1) - Создать.
6. Все элементы диаграммы находятся в левом меню программы (рис. 2). Для создания элемента достаточно потянуть его из меню на страницу.

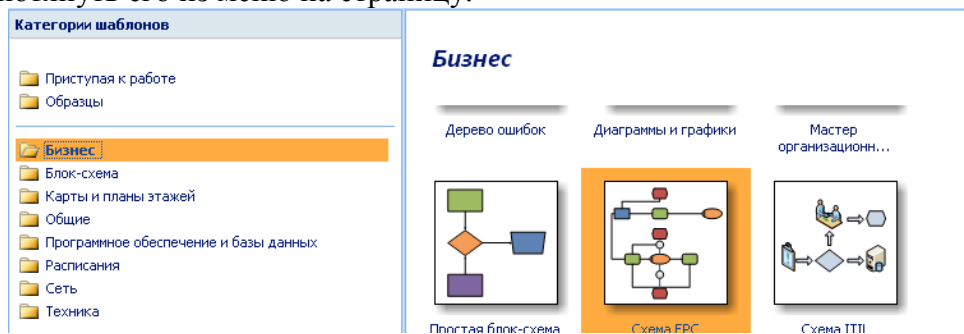


Рисунок 1 – Выбор шаблона Схема ERC

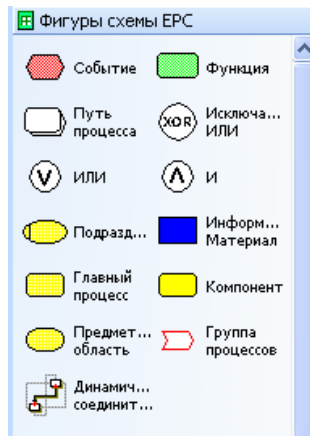
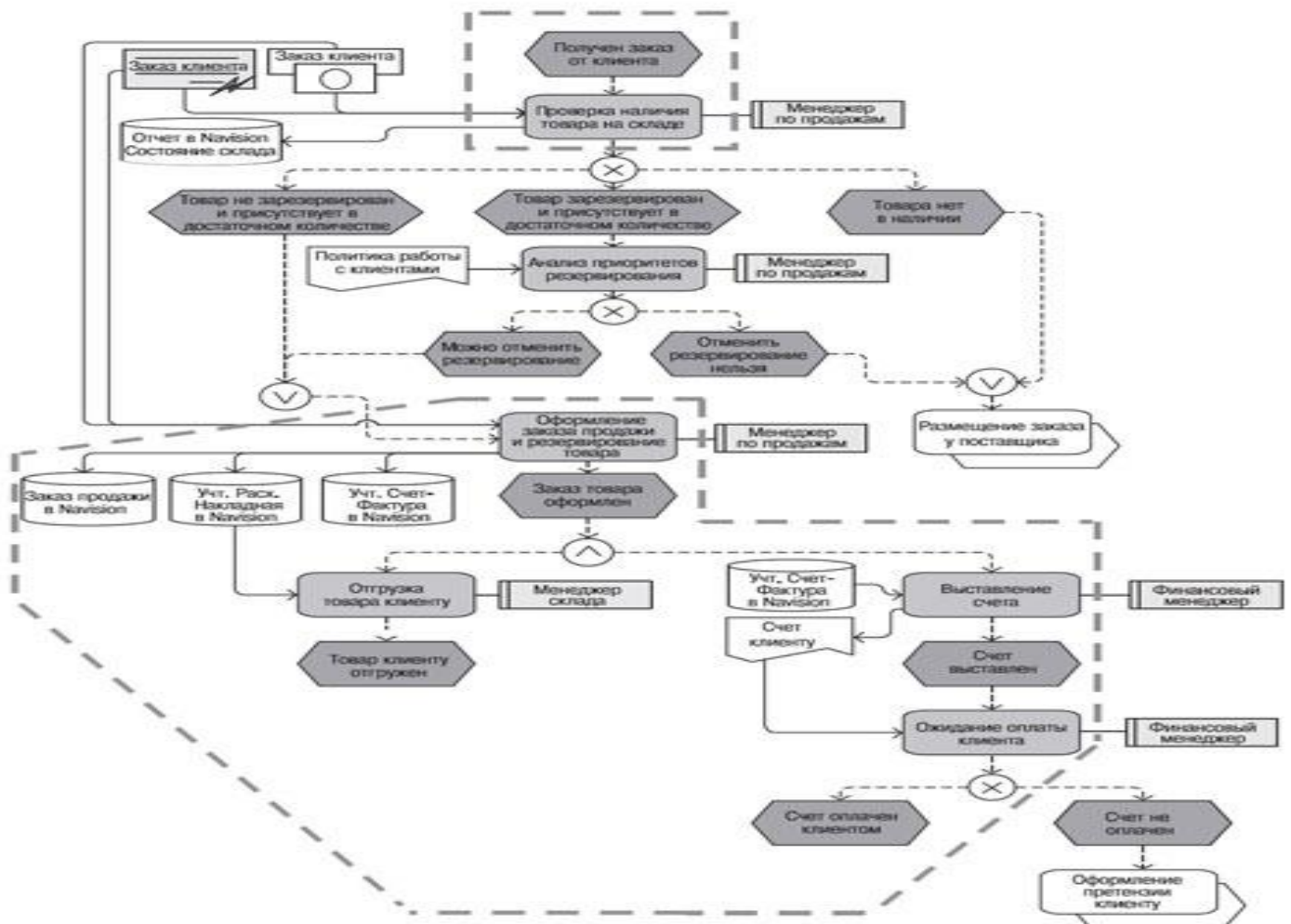


Рисунок 2 – Меню Фигуры

Примечание. Ниже приведен реальный пример цепочки событий от получения заказа до отгрузки товара и оплаты:



Контрольные вопросы:

- 3) Какие логические операторы использованы в вашей диаграмме?
- 4) Каким образом построение диаграммы может помочь при разработке ПО для вашей предметной области?

Сделайте выводы по работе.

Список литературы

Печатные издания

1. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник. Среднее профессиональное образование, профессиональная подготовка / Г.Н. Федорова. – М.: Академия, 2016. – 336 с.

Электронные издания (электронные ресурсы)

1. Гниденко, И. Г. Технология разработки программного обеспечения : учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2019. — 235 с. Текст : электронный // ЭБС Юрайт [сайт]. — Режим доступа: <https://biblio-online.ru/bcode/438444>
2. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности [Электронный ресурс]: учеб. пособие / Г.Н. Федорова. — М. :КУРС : ИНФРА-М, 2019. — 336 с. (Среднее Профессиональное Образование). - Режим доступа: <http://znanium.com/catalog/product/989682>
3. Программное обеспечение компьютерных сетей и web-серверов [Электронный ресурс]: учеб. пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — М. : ИНФРА-М, 2019. — 145 с. - Режим доступа: <http://znanium.com/catalog/product/988332>
4. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем [Электронный ресурс]: учеб. пособие / Л.Г. Гагарина. — М. : ФОРУМ : ИНФРА-М, 2019. — 384 с.- Режим доступа: <http://znanium.com/catalog/product/1003025> (ЭБСZnanium)
5. Технология разработки программного обеспечения [Электронный ресурс]: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. - Режим доступа: <http://znanium.com/catalog/product/1011120> (ЭБСZnanium)
6. Программное обеспечение компьютерных сетей [Электронный ресурс]: учеб. пособие / О.В. Исаченко. — М. : ИНФРА-М, 2019. — 117 с. — (Среднее профессиональное образование). - Режим доступа: <http://znanium.com/catalog/product/989894> (ЭБСZnanium)
7. Плохотников, К.Э. Метод и искусство математического моделирования : курс лекций / К.Э. Плохотников. — 2-е изд., стер. — Москва : ФЛИНТА, 2017. — 519 с. - Текст : электронный. - Режим доступа: <https://new.znanium.com/catalog/product/1034329>
8. Математическое моделирование технических систем [Электронный ресурс]: учебник / В.П. Тарасик. — Минск : Новое знание ; М. : ИНФРА-М, 2019. — 592 с. - Режим доступа: <http://znanium.com/catalog/product/1019246>

Методические издания

1. Игнатенко, Е.С.. Методические указания по выполнению практических работ по МДК 02.01 Технология разработки программного обеспечения.– Нефтеюганск: НИК(филиал) ФГБОУ ВО «ЮГУ», 2019 [Электронный ресурс] Режим доступа: локальная сеть филиала.

Периодические издания

1. Программные продукты и системы [Электронный ресурс]: журнал.- Тверь: — Научно-исследовательский институт «Центрпрограммсистем» Электронно-библиотечная система «Лань» : [сайт]. — Режим доступа: <https://e.lanbook.com/journal/2276>