

Программная инженерия

Анализ и проектирование
программного обеспечения – 1

Вопросы

- Демонстрация описания ТЗ в ЕА
- Проектирование ПО:
 - Общие понятия
 - Подходы к проектированию
 - Применимые стандарты, эскизный проект vs. ADD
 - Типовые архитектуры ПО
 - Zachman framework
- Нотации (старые?):
 - Блок-схема
 - HIPO-диаграмма
 - ERM-диаграмма
 - Data Flow Diagram
 - Псевдо-ЯП

Методы

Технологии

Стандарты

**ПРОЕКТИРОВАНИЕ: ОБЩИЕ
ПОНЯТИЯ**

Определения

- Проектирование ПО — процесс создания проекта ПО
 - Ведется на основе требований к ПО
 - Требования подвергаются анализу
- Цель проектирования:
 - Определение внутренних свойств системы
 - Детализация её внешних (видимых) свойств
- Проектированию подлежат:
 - Архитектура ПО
 - Устройство компонентов ПО
 - Пользовательские интерфейсы

Метод проектирования – 1/3

- Метод проектирования ПО – организованная совокупность процессов создания ряда моделей, описывающих различные аспекты разрабатываемой системы. В метод входят:
- Концепция – теоретическая основа
 - Например: функциональный подход, модульный подход, структурный подход, объектно-ориентированный подход

Метод проектирования – 2/3

- Нотация – способ представления моделей (обычно графические)
 - Блок-схема
 - ER-диаграмма
 - UML-диаграммы (ООП): диаграммы вариантов использования, диаграммы классов
 - Для структурного проектирования: диаграммы потоков данных, и диаграммы «сущность – СВЯЗЬ»

Метод проектирования – 3/3

- Процедуры – определяют практическое применение метода
 - Последовательность и правила построения моделей
 - Критерии, используемые для оценки результатов

Технология проектирования

- Методы проектирования реализуются через конкретные:
 - Технологии и поддерживающие их методики
 - Стандарты
 - Инструментальные средства
- Технология проектирования – совокупность технологических операций проектирования в их последовательности и взаимосвязи, приводящая к разработке проекта ПО

Требования к технологии – 1/2

- Общие формальные:
 - Соответствие стандарту (стандартам)
 - Гарантированное достижение целей разработки при соблюдении ограничений:
 - Бюджет
 - Заданное качество
 - В установленное время
- Организационные для заказчика:
 - Минимальное время до получения рабочего прототипа или отдельной рабочей подсистемы

Требования к технологии – 2/2

- Организационные для разработчика:
 - Разбиение проекта на части, доступные для [независимой] реализации небольшими группами разработчиков (3-7 человек) с последующей интеграцией
 - Независимость проекта от реализации (ЯП, СУБД)
 - Наличие поддерживающих CASE средств

Подходы к проектированию – 1/2

- Модульный
 - Разбиение ПС на модули – составные части, которые могут быть независимо написаны и отлажены
- Структурный
 - Представление ПС в виде (иерархической) многоуровневой системы компонентов
- Функциональный
 - Разделение на части, реализующие отдельные функции
- $OO = 1 + 2 + 3$

Подходы к проектированию – 2/2

- **Сверху вниз:**
 - Начинают с верхнего уровня (обычно включающего внешние связи всей проектируемой системы), разбивают систему на части
 - Каждую часть разбивают далее
- **Снизу вверх:**
 - Сначала описывают наиболее мелкие элементы системы
 - Из них выстраивают более крупные

Стандарты

- ISO/IEC/IEEE 42010:2011 Systems and software engineering. Architecture description (Системная и программная инженерия. Описание архитектуры)
- ГОСТ 2.103-68 ЕСКД. Стадии разработки
- ГОСТ 2.119-73 ЕСКД. Эскизный проект
- ГОСТ 2.120-73 ЕСКД. Технический проект

Типы архитектур – 1/2

- Архитектуры потоков данных:
 - Последовательные пакеты
 - Каналы и фильтры
- Независимые компоненты:
 - Параллельные взаимодействующие процессы
 - Клиент-серверные системы
 - Системы, управляемые событиями
- Виртуальные машины:
 - Интерпретаторы
 - Системы, основанные на правилах

Типы архитектур – 2/2

- Репозиторные архитектуры:
 - Базы данных
 - Гипертекстовые системы
 - Доски объявлений
- Уровневые архитектуры

Zachman Framework – 1/4

- Модель описания архитектуры системы, предоставляющая формальный структурированный способ описания. Представляет собой матрицу:
 - 6 столбцов – «стороны» представления системы
 - 6 строк – уровни представления системы
- Обычно при описании используют в качестве примера проектирование здания







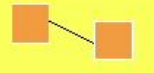























Zachman Framework: Views – 2/4

1. Planner's View (Scope)
 - Представление инвестора или планирующего деятельность для получения общего описания и оценки стоимости и связи с внешним миром
2. Owner's View (Enterprise or Business Model)
 - Представление владельца, использующего систему в повседневной деятельности
3. Designer's View (Information Systems Model)
 - Представление разработчика архитектуры, переводящего требования в архитектурные решения
4. Builder's View (Technology Model)
 - Описание конкретных способов реализации архитектуры
5. Subcontractor View (Detailed Specifications)
 - Подробные описания, достаточные для реализации и/или покупки и настройки конкретных модулей
6. Actual System View
 - Сама реализованная система

Zachman Framework: «стороны» – 3/4

- What
 - Описание данных
- How
 - Описание функций
- Where
 - Описание сети (развертывания)
- Who
 - Описание персонала / пользователей
- When
 - Описание временных зависимостей
- Why
 - Цели и задачи

Zachman Framework – 4/4

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE {contextual} Planner	List of things important to the business  Entity = Class of business things	List of processes the business performs  Process = Class of business process	List of locations in which the business operates  Node = Major business locations	List of organisations important to the business  People = Major business unit	List of event cycles significant to the business  Time = Major Business Event Cycle	List of business goals/strategies  End/Mean = Major Business Goal/Strategy
BUSINESS MODEL {Conceptual} Owner	e.g., Semantic Model  Entity = Business Entity Relationship = Business	e.g., Business Process Model  Process = Businessess I/O = Business Resource	e.g., Business Logistics System  Node = Business Location Link = Business Linkage	e.g., Workflow Model  People = Organisation unit Work = Work Product	e.g., Master Schedule  Time = Business Event Cycle = Business Cycle	Business Plan  End = Business Objective Means = Business Strategy
SYSTEM MODEL {Logical} Designer	e.g., Logical Data Model  Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture  Process = Application Function I/O = User Views	e.g., Distributed System Model  Node = I/S Function Relationship = Line Characteristics	e.g., Human Interface Architecture  People = Role Work = Deliverable	e.g., Processing Structure  Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model  End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL {Physical} Builder	e.g., Physical Data Model  Entity = Segment/Table Relationship = Pointer/key	e.g., System Design  Process = Computer Function I/O = Data Elements/sets	e.g., Technology Architecture  Node = H/w /System s/w Relationship = Line Specifications	e.g., Presentation Architecture  People = User Work = Screen Formats	e.g., Control Structure  Time = Execute Cycle = Component Cycle	e.g., Rule Design  End = Condition Means = Action
DETAILED REPRESENTATIONS {Out-of-context} Subcontractor	e.g., Data Definition  Entity = Field Relationship = Address	e.g., Program  Process = Language Statement I/O = Control Block	e.g., Network Architecture  Node = Address Link = Protocol	e.g., Security Architecture  People = Identity Work = Job	e.g., Timing Definition  Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification  End = Sub-condition Means = step
FUNCTIONING ENTERPRISE	e.g DATA	e.g FUNCTION	e.g NETWORK	e.g ORGANISATION	e.g SCHEDULE	e.g STRATEGY

Блок-схема

HIPO-диаграмма

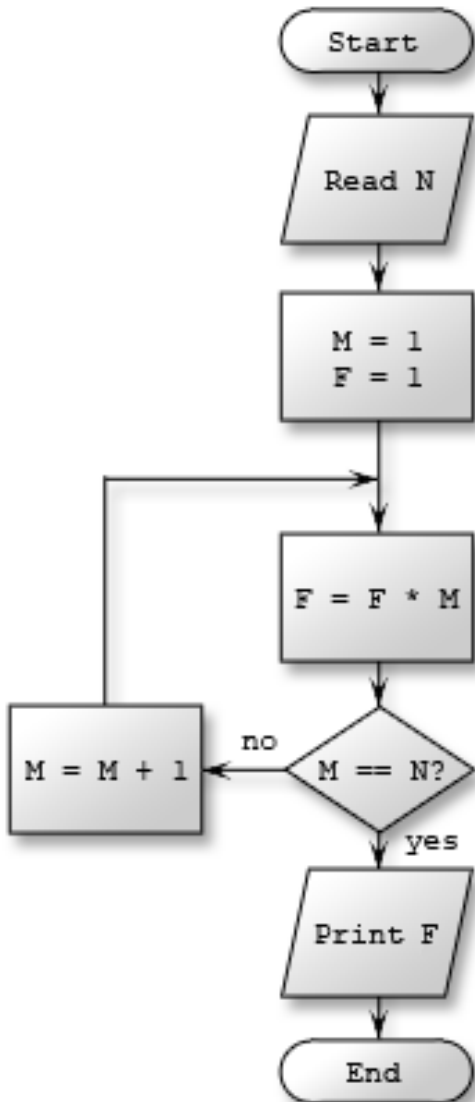
ERM-диаграмма

DFD






Псевдо-ЯП

НОТАЦИИ

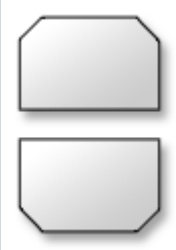


Нотация: блок-схема – 1/4



Нотация: блок-схема – 2/4

Блок	Описание
	Начало-конец (пуск-остановка)
	Блок действия Набор последовательно и неразрывно выполняемых операций
	Логический блок (условие) Условие выбора дальнейшего пути (может делить на 2 и более путей)
	Процедурный блок
	Ввод-вывод данных

Нотация: блок-схема – 3/4

Блок	Описание
	Начало и конец цикла
	Соединитель Разрывы линий на одном листе Переходы на другие листы
	Комментарий

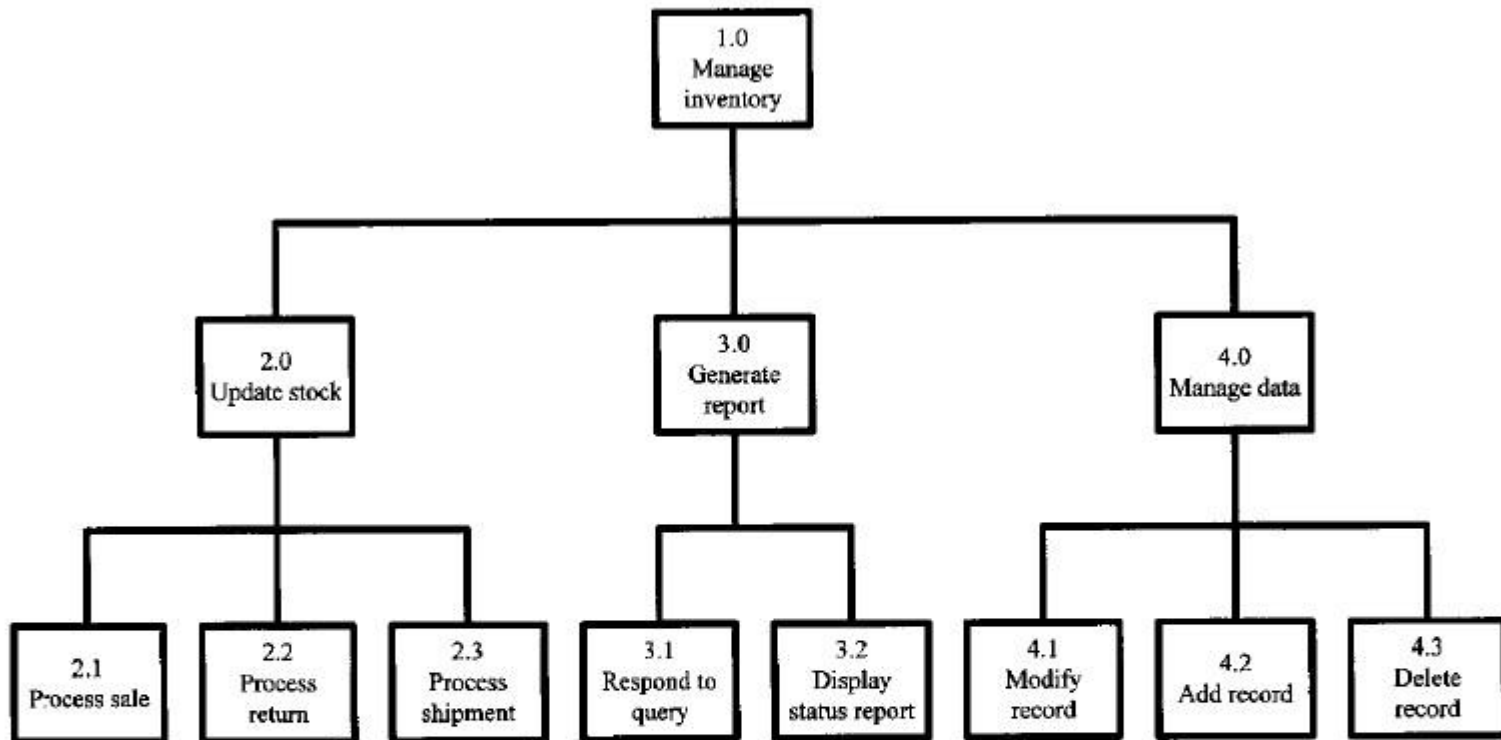
Нотация: блок-схема – 4/4

- Преимущества:
 - Распространенность
 - Простота понимания
- Недостатки:
 - Сложные алгоритмы выглядят сложно
 - Сложные отношения между элементами ПО сложно отобразить
 - Применимо только к нижнему уровню

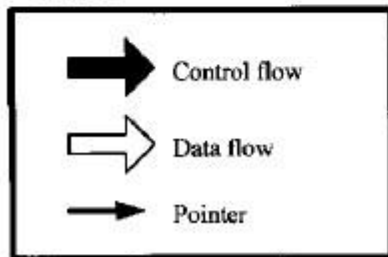
Нотация: HIPO – 1/3

- Hierarchical input process output (HIPO) — технология проектирования и документирования. Состоит из:
 - Диаграммы иерархии
 - IPO – диаграммы процессов с описанием входной и выходной информации
- Недостатки:
 - Алгоритмы тяжело представить
 - Современные программы не всегда иерархичны
- Преимущества:
 - Описывает: иерархию, IO, передачи управления
 - Ностальгия!

Нотация: HIPO – 2/3



Legend



Нотация: HIPO – 3/3

Author: W. S. Davis

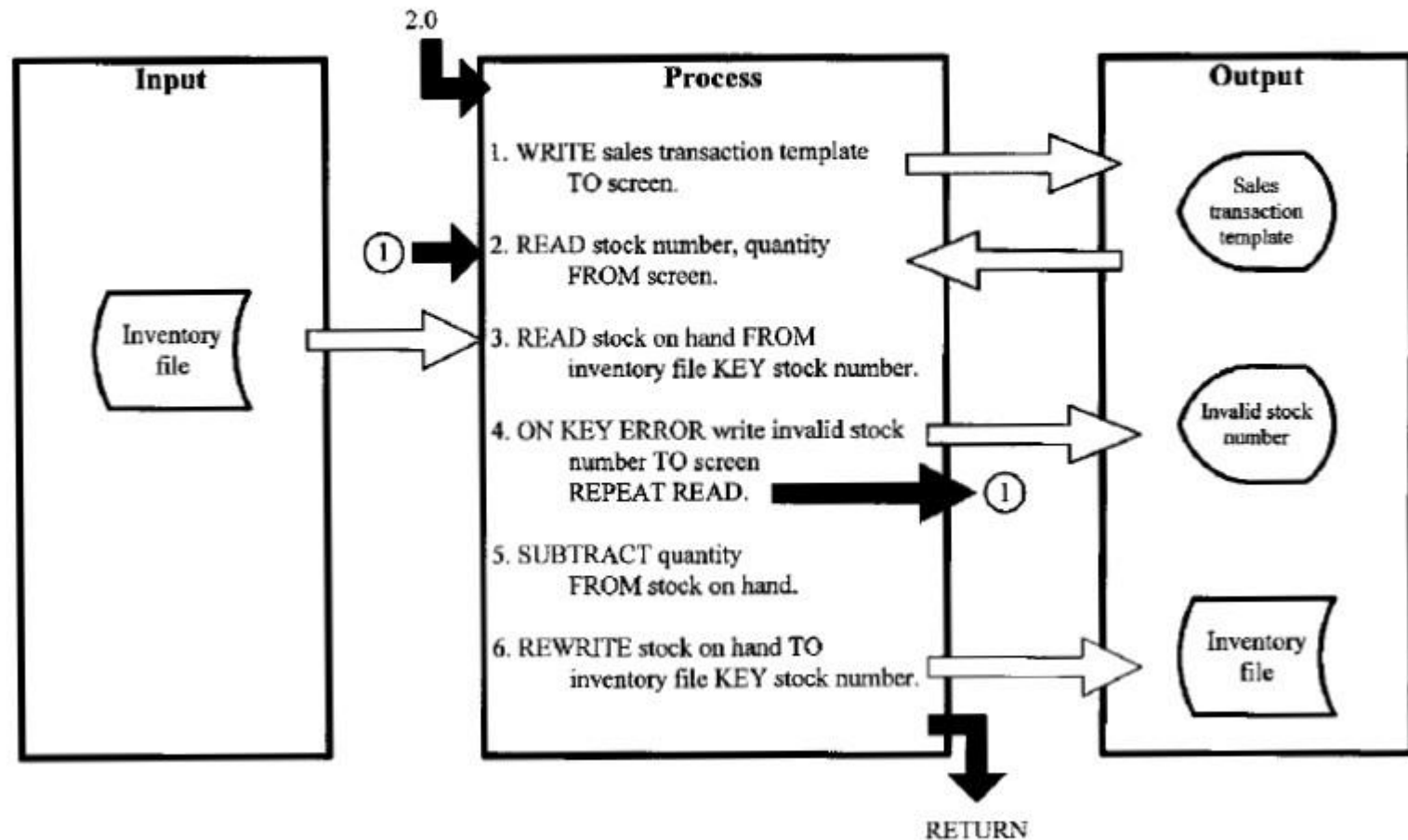
Program: Inventory

Date: _____

Diagram: 2.1 (Detailed)

Module name: Process sale

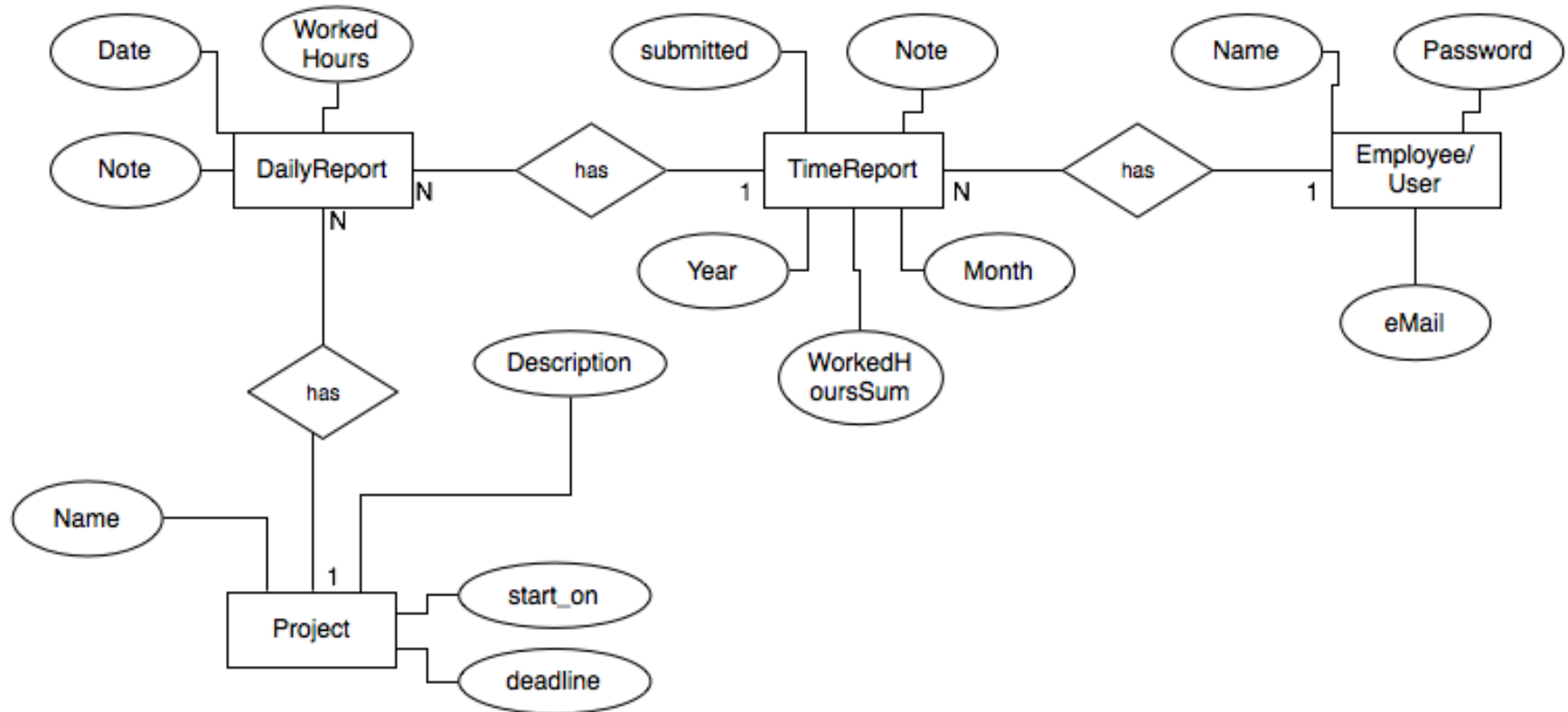
Page _____ of _____



Нотация: ER-диаграмма – 1/2

- Модель сущность-связь (ER-модель – entity-relationship) — модель данных, позволяющая описывать концептуальные схемы предметной области
- Была создана для проектирования БД
- Нотация Питера Чена:
 - Сущности – прямоугольники
 - Отношения – ромбы
 - Линия – отношение между сущностями, если отношение не является обязательным, то линия пунктирная
 - Овалы – атрибуты отношения или сущности

Нотация: ER-диаграмма – 2/2



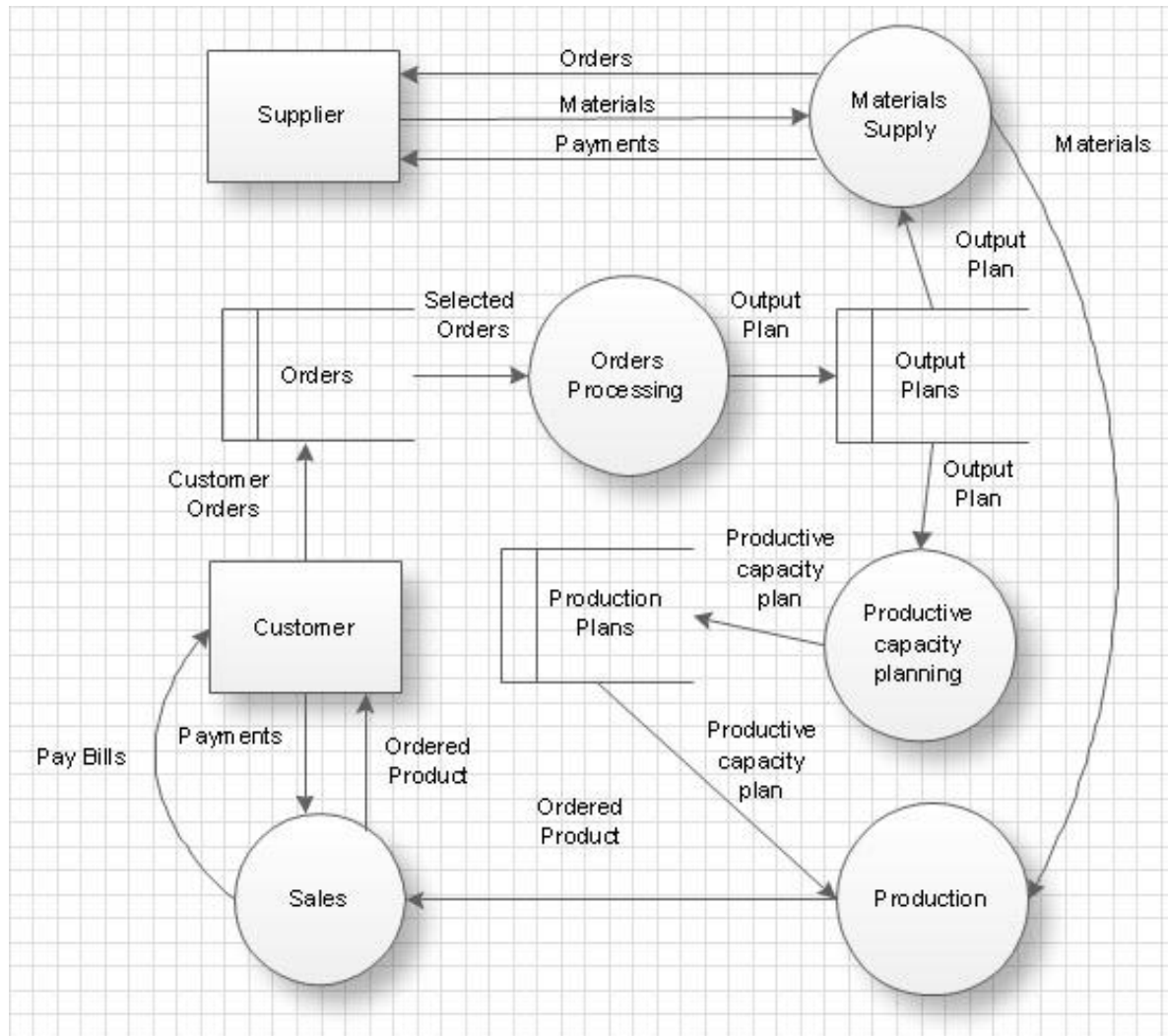
Нотация: Data Flow диаграмма – 1/3

- DFD (Data Flow Diagrams) — диаграммы потоков данных
- Преимущества:
 - Иерархическая
 - Верхний уровень иерархии описывает внешние интерфейсы системы
 - Поддерживает понятие подсистемы
- Недостатки:
 - Сложное поведение (не обработку потока данных) представить затруднительно

Нотация: Data Flow диаграмма – 2/3

- Элементы:
 - Внешняя сущность (External Entity)
 - Системы и подсистемы (Process)
 - Например: подразделение, программа
 - Поток данных

Нотация: Data Flow диаграмма – 3/3



Нотация: Псевдо ЯП – 1/2

- Алгоритм описывается на ЕЯ с использованием для структурирования элементов ЯП
 - Например: PDL (Program Design Language)
- Преимущества:
 - Понятно разработчикам
 - Компактнее и выразительнее чем блок-схема
 - Легко конвертируется в код:
 - ЯП остаются как есть, ЕЯ куски становятся комментариями
- Недостатки:
 - Непонятно не-программистам
 - Применимо только на нижнем уровне

Нотация: Псевдо ЯП – 2/2

```
Boolean купитьКолбасы()  
{  
    if( яйца есть? )  
    then  
        купить( 10 )  
    else  
        купить( 1 )  
    endif  
    return TRUE  
}
```