Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Tehnologii CASE”

Экзаменационная работа

Вариант 2

Проверил: Гладей Анатолий

Выполнил: Маруневич Николай

Кишинев 2022

## 1. Методологии-предшественники UML: Сети Петри, диаграммы Харела, DFD, метод Буча. (20 p.)

### 1.1 Сети Петри

Появились в 1962 году и предназначались прежде всего для изучения свойств асинхронных, параллельных или же недетерминированных систем. Данный тип диаграмм послужил базой для других типов диаграмм по типу блок-схем.

Данные диаграммы представляют собой двудольный граф, который состоит из из вершин(называемых *“позиция”*) и мостов между ними(называемых *“переход”*). Вершины содержат токены исполнения, которые забираются/добавляются мостами в зависимости от типа связи с вершиной, также мосты имеют ещё параметр, который именуется *“вес”*, который означает кратность данного моста*.*

Смотря с математической точки зрения, сети Петри очень хорошо изучены и при хорошем понимании их начальной разметки и топологии можно очень легко определить тип данной сети(*живая / безопасная*), вероятные разметки исходящие из данной. Данные свойства помогают облегчить аналитический анализ программ на присутствие каких-либо багов

### 1.2 DFD(Data Flow Diagram)

Представитель старожилов диаграмм потоков данных, который предлагает пользователю всего лишь три типа объектов для использования. Данными типами являются:

* Активности
* Хранилища данных
* Внешняя сущность

Для связи данных объектов используются стрелки, которые показывают направления потока данных между объектами. Важно, каждый объект может представлять собой отдельную диаграмму, которая описывает более точное поведение объекта при поступлении в него каких-либо данных. Также, ещё одним важным аспектом данного типа диаграмм является то, что она не задает порядок действий, а обработку информации может выполнить любая из активностей, с учетом того, что она имеет всё необходимое для совершения данного действия.

### 1.3 Диаграммы Харела

Данный тип диаграмм является самым популярным в научных кругах и стал одним из кирпичиков фундамента множества методологий, и почти без изменений состоит в UML. При помощи данных диаграмм можно реализовать визуализацию состояния и переходы *“машин”*, определять подсистемы, которые работают параллельно независимо от друг друга, вложенные состояния и многое другое.

Диаграммы Харела очень удобны при спецификации работы устройств, имеющих какие-либо состояния и имеют возможность обрабатывать разные события, поведение обработки определяется исходя из текущего состояния устройства. В качестве примера таких устройств можно использовать как обычные часы, так и сетевой сокет, деятельность которых с легкостью может быть представлена в таком виде.

Данные диаграммы характеризуются простой и четкой семантикой, что позволяет выполнять генерацию кода по ним.

### 1.4 Метод Буча

Данный метод был описан в книге одного из отцов основателей языка UML. Состоит из следующих типов диаграмм:

* Классов
* Состояний
* Переходов
* Объектов
* Взаимодействий
* Модулей
* Процессов

Каждая из диаграмм имеет свой стиль. Стили некоторых перешли в UML почти не претерпев изменений.

## 2. Классификация визуальных языков (15 p.)

Визуальные языки могут быть классифицированы по следующим критериям:

* Язык основывающийся на объектах, где среда разработки представляет графические или же символьные элементы, подвергаемые манипуляции в интерактивном стиле с учетом установленных правил
* Конструкторы формы/интерфейса, в которых пользователь указателем выставляет выбранные части интерфейса и настраивает их свойства исходя из которых может быть сгенерирован код.
* Язык диаграмм/схем, в которых представлены только фигуры и линии комбинация которых представляет их взаимоотношения.

Каждая из данных категорий может быть использована в различных сферах деятельности и аспектах. На сегодняшний день язык UML содержит в себе около 14 видов различных диаграмм, которые хорошо отличаются между собой.

## 3. Основные этапы развития CASE-инструментов (15 p.)

Развитие CASE-инструментов можно разделить на следующие этапы:

1. ***Первое поколение*** —> имеют средства, которые обеспечивают увеличение производительности труда и качества проектирования на определенных этапах разработки Информационных Систем.
2. ***Второе поколение*** —> создается интегрированная среда которая позволяет комплексно автоматизировать процесс проектирования Информационных Систем. На данном этапе инструменты данной категории объединялись одной методологией проектирования. Они имели общее программное, математическое, лингвистическое, организационное обеспечение.
3. ***Третье поколение*** —> эволюция среды в так называемую *“общую систему”*, которая представляла собой стандартную интерпретируемые систему, которая представляла изоморфный класс систем, по отношению к другим классам подобных технологий

## 4. Структура визуального языка (синтаксис, семантика, прагматика) (20 p.)

**Синтаксис**:

1. Абстрактный синтаксис
   1. BNF
   2. Метамодели
2. Конкретный синтаксис
   1. Графические грамматики
   2. Формальное описание
   3. Редактор форм, скрипты
   4. Неформальные описания
3. Служебный синтаксис
   1. Модели
   2. Диаграммы

**Семантика**

1. Генерация
2. Интерпретация
3. Преобразование моделей

**Прагматика**

1. Описывает взаимосвязь пользователей и языка
2. Научные подходы
3. Набор неформальных рекомендаций

## 5. Основные принципы предметно-ориентированного моделирования, сравнение с другими подходами (15 p.)

### 5.1 Ограничение связи

Представляется в виде использования нескольких моделей на разных уровнях проекта. Использование такого подхода уменьшает различные связи между моделями и позволяет избавиться от большой сложности и запутанности модели/кода.

### 5.2 Целостность

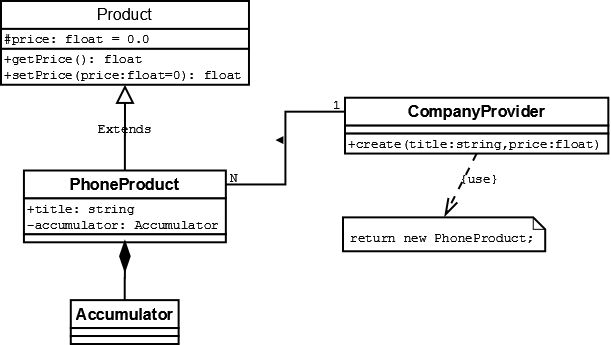
Зачастую в больших проекта присутствует манера дробления модели на более мелкие, что не всегда идёт на пользу и ускорение процесса создания финального продукта

### 5.3 Взаимосвязь

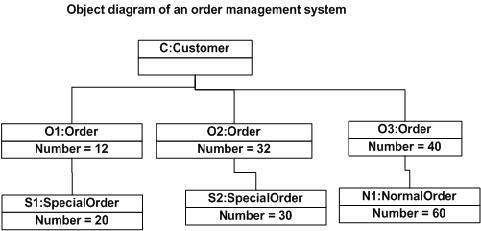
Позволяет упростить и ускорить процесс разработки и конечной сборки финального продукта, так как избавляет такой фактор как незнание одной команды о сущности другой модели при работе в большой команде, разделенной на подгруппы

## 6. Диаграммы UML: диаграммы классов, объектов, пакетов, компонентов, составных структур, развёртывания(15 p.)

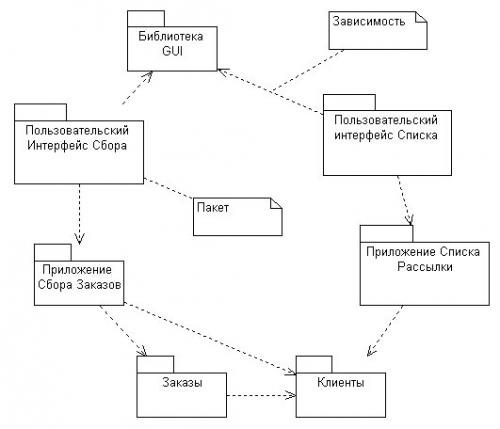
### 6.1 Диаграмма Классов



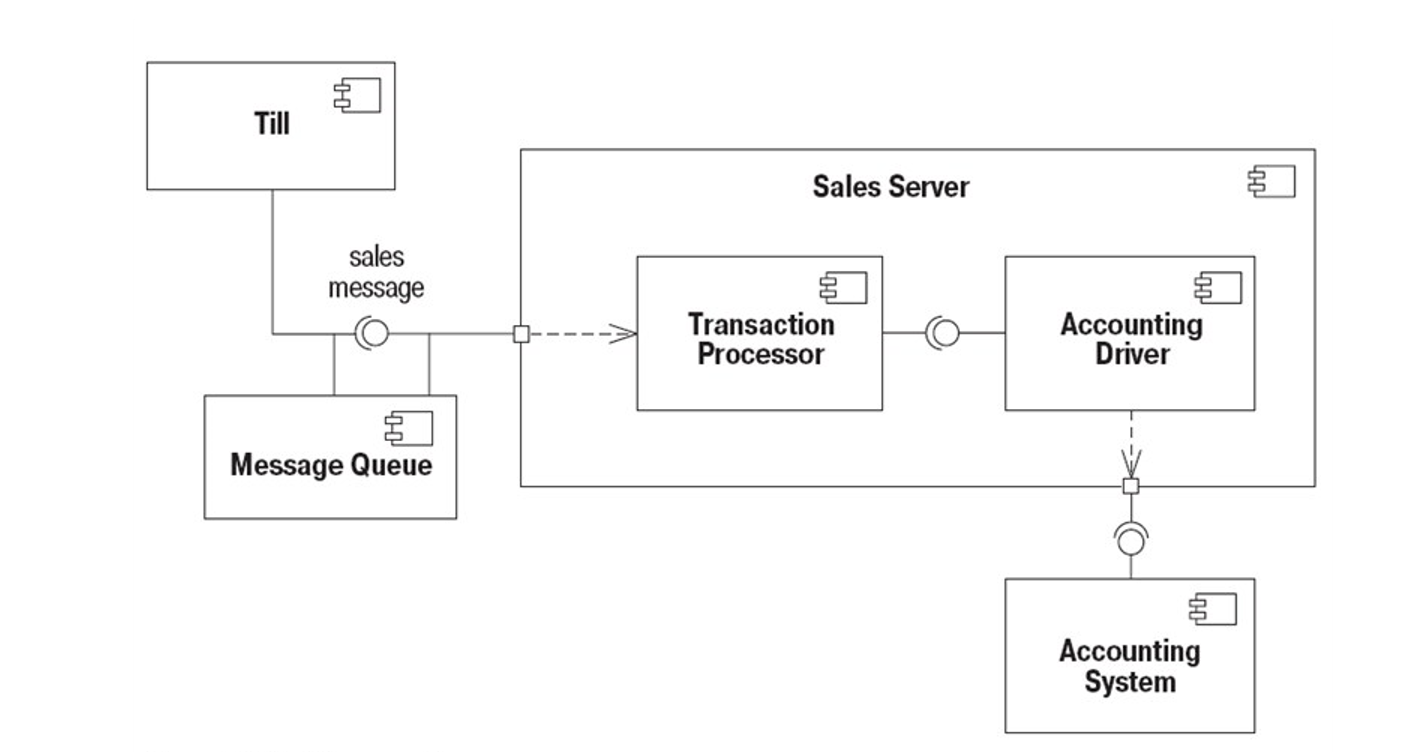
### 6.2 Диаграмма Объектов



### 6.3 Диаграмма Пакетов



### 6.4 Диаграмма Компонентов



### 6.5 Диаграмма Развертывания

