

Технологии программирования  
2 семестр

Автор

Весна 2020

---

Это конспект лекций читаемых в МФТИ Старичковым Н.. Название курса "Технологическое Программирование

# Оглавление

|          |                                   |          |
|----------|-----------------------------------|----------|
| <b>1</b> | <b>Проектирование ПО</b>          | <b>1</b> |
| 1.1      | Этапы проектирования. (1я часть)  | 1        |
| 1.1.1    | Формирование требований           | 1        |
| 1.1.2    | Разработка концепции              | 2        |
| 1.1.3    | Техническое задание               | 2        |
| 1.1.4    | Эскизный проект                   | 2        |
| 1.1.5    | Технический проект                | 3        |
| 1.1.6    | Рабочая документация              | 3        |
| 1.1.7    | Ввод в действие                   | 3        |
| 1.1.8    | Сопровождение системы             | 3        |
| 1.2      | UML (2я часть)                    | 4        |
| 1.2.1    | Диаграмма вариантов использования | 4        |
| 1.2.2    | Диаграмма классов                 | 5        |
| 1.2.3    | Диаграмма последовательности      | 6        |
| 1.2.4    | Диаграмма состояний               | 7        |
| 1.2.5    | Диаграмма деятельности            | 9        |



# Глава 1

## Проектирование ПО

### 1.1 Этапы проектирования. (1я часть)

1. Формирование требований
2. Разработка концепции
3. Техническое задание
4. Эскизный проект
5. Технический проект
6. Разработка документации
7. Поставка / ввод в действие
8. Сопровождение

#### 1.1.1 Формирование требований

1. Общение с клиентом
  2. Общение с пользователем
  3. Анализ прикладной части
  4. Формирование оценок требуемой производительности
  5. Обоснование объекта
  6. Исследование необходимости проекта (какие проблемы решает).
  7. Формирование требований пользователя
- подготовка отчета по этапу

### 1.1.2 Разработка концепции

1. Изучение объекта автоматизации
  2. Проведение необходимых НИР (научно исследовательских работ)
  3. Разработка вариантов концепции
  4. Выбор формата (это веб ресурс / приложение)
  5. Целевое оборудование
  6. Построение высокоуровневой архитектуры системы
  7. Выбор / Разработка новых алгоритмов / технологий
- подготовка отчета по этапу

### 1.1.3 Техническое задание

Основное отличие от документации в том что тут находятся данные о требованиях предоставляемых разработчику (например должно работать с 90% железа), которые пользователь может не видеть. А документация как раз доступна для пользователя.

1. Описание системы
2. Описание функциональности
3. Описание сценариев использования  
Например, пользователь обычно либо только выгружает видео или только стримит, и ему не нужно выделять ресурсов для одновременности этих процессов.
4. Условия сдачи

### 1.1.4 Эскизный проект

1. Разработка прототипов частей системы
  2. Оценка производительности и качества
  3. Изменение прототипов
- Часто это MVP (minimal viable product)
  - Иногда это система с базовой функциональностью
  - Иногда с урезанным проектом
  - Иногда менее производительная
- подготовка отчета по этапу

### **1.1.5 Технический проект**

1. Разработка частей системы
2. Разработка документации
3. Разработка заданий на проектирование и реализацию основных частей
4. Тестирование
5. Оценка качеств и производительности

### **1.1.6 Рабочая документация**

1. Сценарии использования
2. Описание логики работы
3. Описание производительности
4. Примеры использования
5. Обучающие мероприятия

### **1.1.7 Ввод в действие**

1. Подготовка объекта автоматизации
2. Подготовка персонала
3. Комплектация системы поставляемыми изделиями
4. Проведение предварительных испытаний
5. Опытная эксплуатация
6. Приемочные испытания

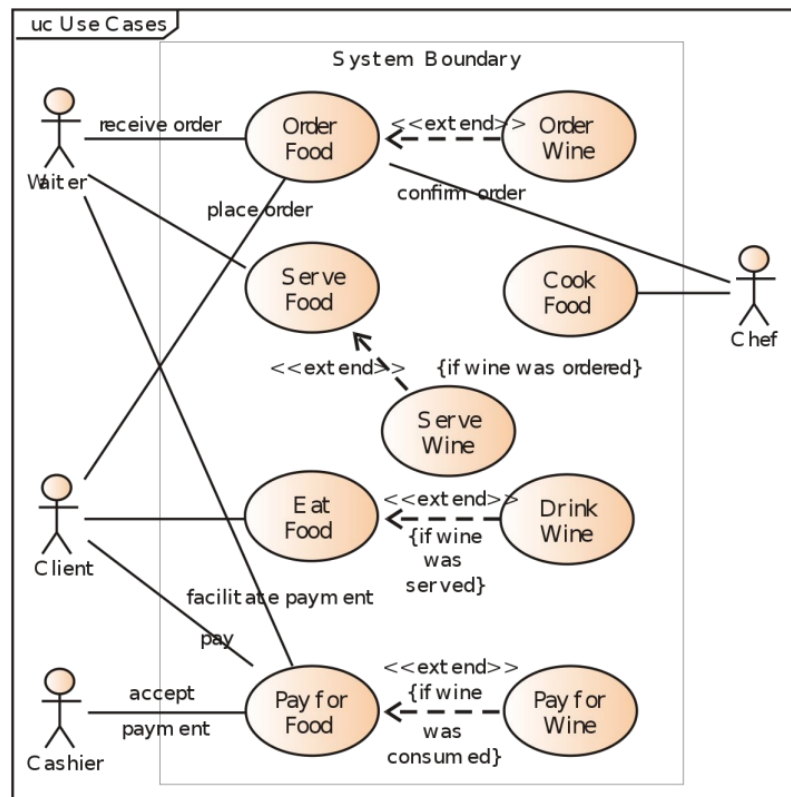
### **1.1.8 Сопровождение системы**

1. Гарантийные обязательства
2. Послегарантийное обслуживание

## 1.2 UML (2я часть)

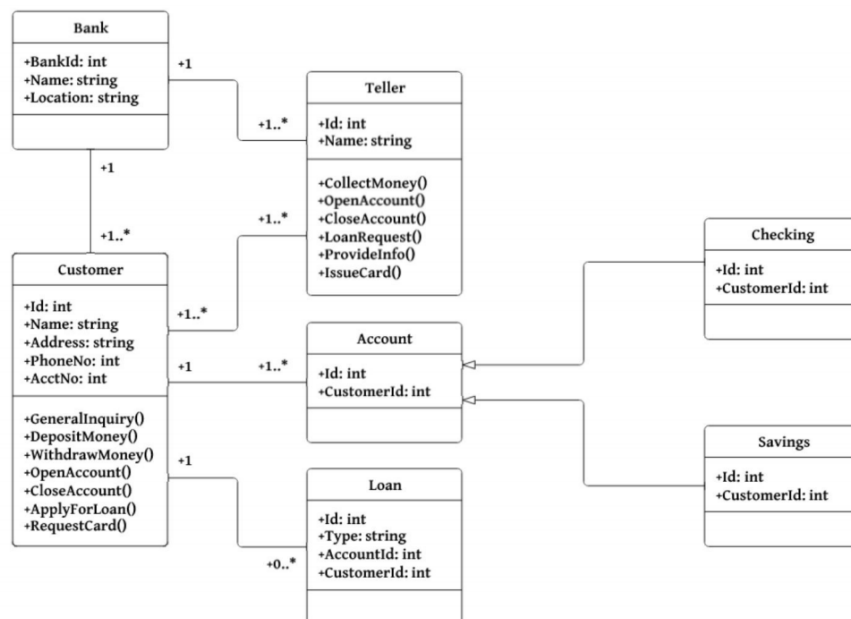
UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. In this article, we will give you detailed ideas about what is UML, the history of UML and a description of each UML diagram type, along with UML examples.

### 1.2.1 Диаграмма вариантов использования





## 1.2.2 Диаграмма классов



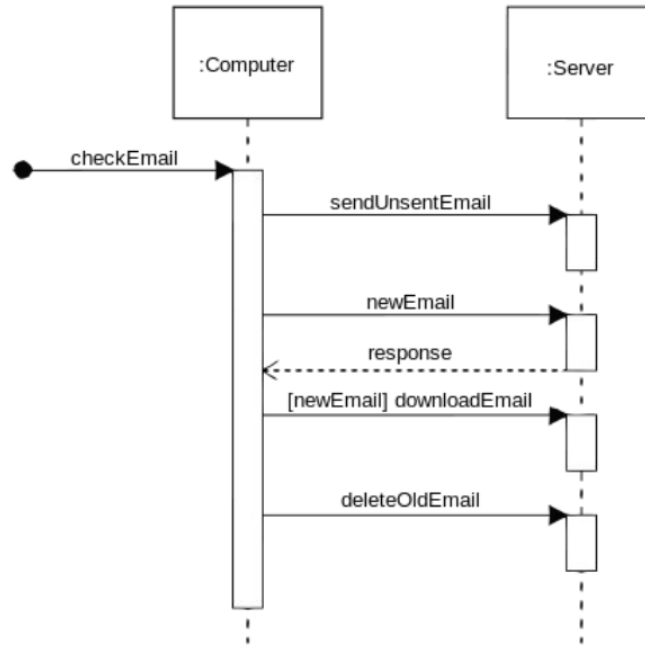
Есть разные связи:

1 (Например король Артур и его лошадь) когда один объект так же перестает существовать

1-1 Лошадь и всадник, когда они существуют по отдельности

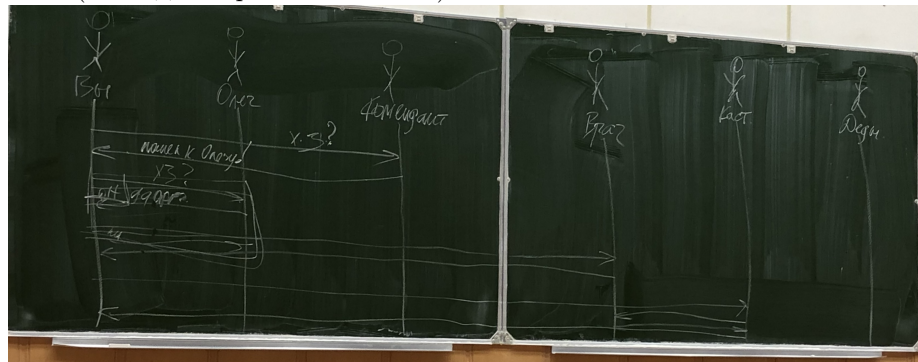
Так, например, в базе данных при удалении одного элемента, элементы, связанные с ним связью "1" так же будут не действительны.

## 1.2.3 Диаграмма последовательности



Рассмотрим на примере заселения. В процессе участвуют Мы, Олег (поселяющийся), Врач, кастелянша, деканат. Сначала мы идем к камендантше, затем идем к Олегу, он дает "а проге". После идем к врачу, потом уже к кастелянше.

Причем подключаются блоки в разных случаях: Сказал ли Олег Да или Нет? (Блок договориться с Олегом)



## 1.2.4 Диаграмма состояний

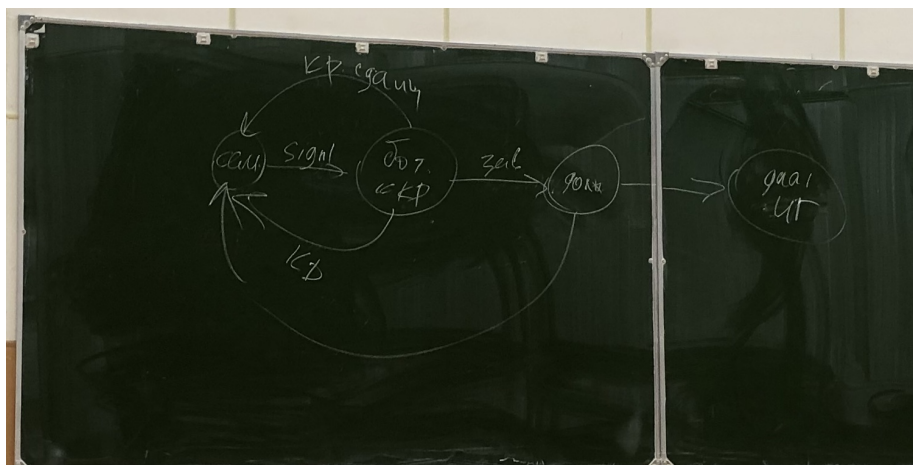
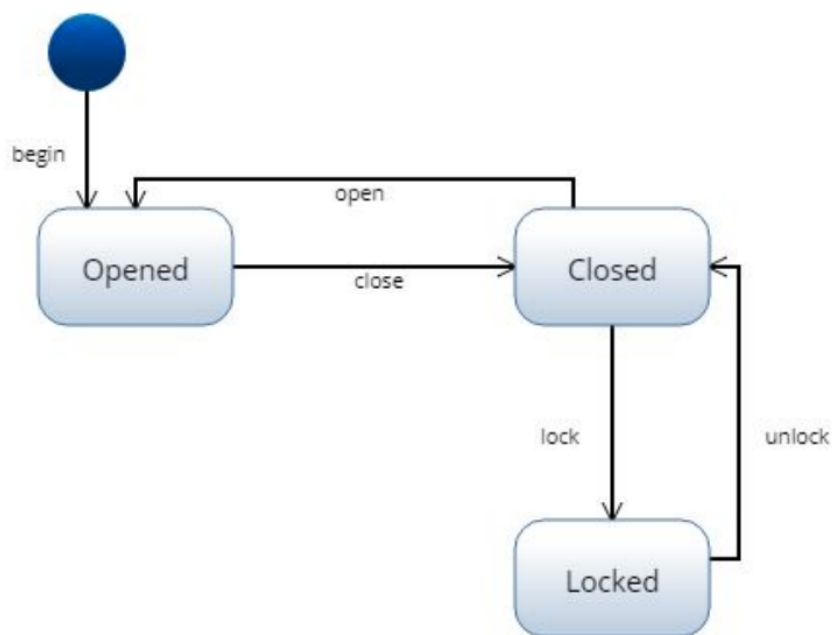
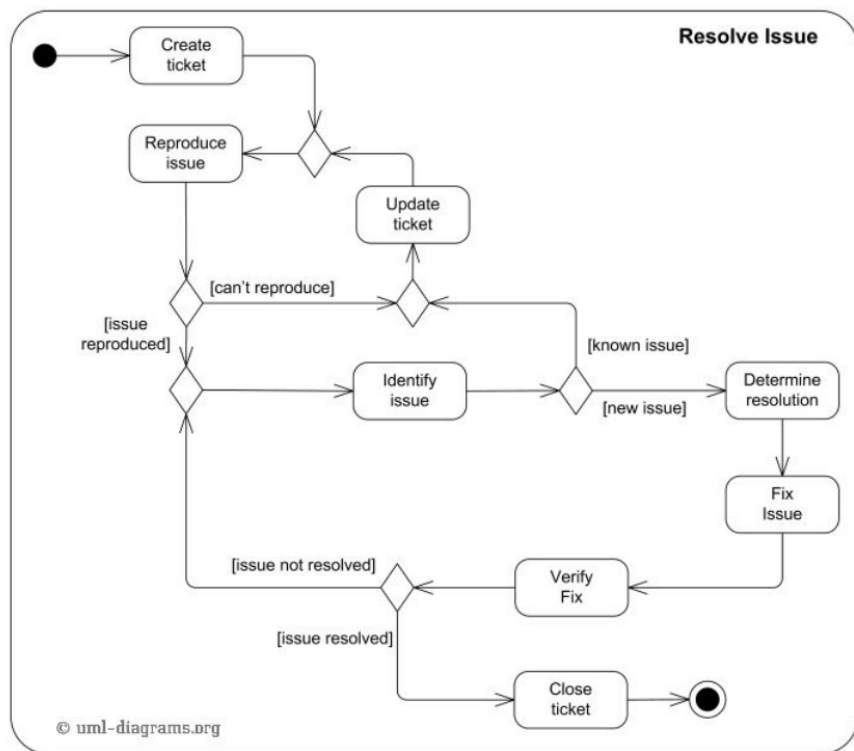


Диаграмма для студента: Студент находится в состоянии "учусь иногда"

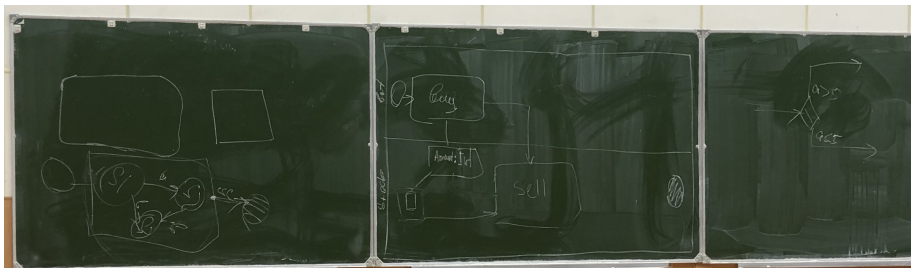
переходит к состоянию "ботаю кр дальше если кр не сдана то к состоянию "должник" и так далее...

Так с экраном. Если вы нажимаете Esc проваливаетесь на рабочий стол, дальше если открываете приложение проваливаетесь еще куда-то и т. д. Еще когда мы открываем на телефоне карточки для платежа, яркость телефона выкручивается на максимум. Иногда телефон забывает вернуть яркость обратно, И ОН ВЫЖИГАЕТ ГЛАЗА) И нужно прописывать при переходе это.

## 1.2.5 Диаграмма деятельности



Описание порядка набора действий на формальном языке с указанием что кто вызывает, по которой можно все восстановить.



#операция - с закругленными, объект - прямоугольник с острыми углами  
 например у bot 'а операция buy передает объект "amount int" в операцию

биржи(snoxs) выполняя ее

#терминальное состояние – состояние конца операции для которой рисуется диаграмма

Еще есть условия (Например Ammount > 5/ <5...)

#условия рисуются ромбиком



Еще есть распаралеливание и при завершении одного потока завершается только один поток. Потоки можно сливать. Например я хочу получить визу. Система паралельно проверяет паспорт, проверяет банковский счет, и последняя, сразу завершаемая: передать в интерпол что поступил запрос. Дальше все сливается и виза дается/или нет.