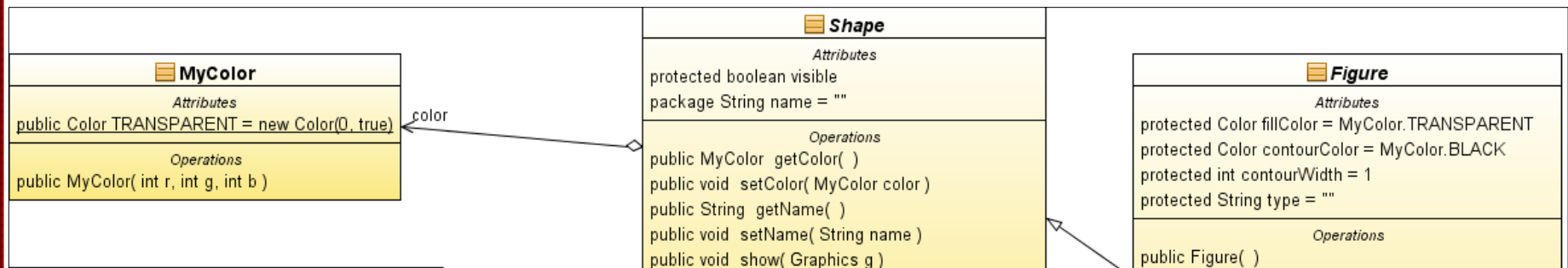


# Спецификация на изискванията и реализация на учебни проекти с Java™



Траян Илиев

IPT – Intellectual Products & Technologies  
e-mail: [tiliev@iproduct.org](mailto:tiliev@iproduct.org)  
web: <http://www.iproduct.org>

Oracle®, Java™ and EJB™ are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Oracle®, Java™ и EJB™ са търговски марки на Oracle и/или негови подразделения. Всички други търговски марки са собственост на техните притежатели.

## Съдържание

1. Спецификация на изискванията
2. Use Case модел на изискванията към проекта
3. Model-View-Controller (MVC) архитектура на софтуерните приложения
4. Ентити (модел на данните), контролни и презентационни класове
5. Структура на учебните проекти - първи срок

## Изисквания към софтуерния продукт

- Изискване (Requirement) [Young, 01] =
  - отделна документирана нужда относно това, което конкретен продукт или услуга трябва да бъде или да прави
  - изявление, което идентифицира необходим атрибут, способност, характеристика или качество на една система, за да бъде тя полезна за потребителя

## Класификация на видовете изисквания

- Изисквания към продукта срещу изисквания към процеса на разработка
  - **бизнес изисквания** – какво трябва да бъде направено или изпълнено в бизнес термини
  - **изисквания към продукта** – описват системата или продукта като един възможен начин за удовлетворяване на бизнес изискванията
  - **изисквания към процеса** – описват процеса, който разработващата организация трябва да следва
- **Функционални** (способности) срещу **нефункционални** (ограничения) изисквания

## Критерии за добре специфицирани изисквания

- Кохерентност
- Пълнота
- Съгласуваност
- Коректност
- Актуалност
- Външна наблюдаемост
- Полезност
- Еднозначност
- Задължителност
- Верифицируемост

## Управление на изискванията [Young]

- Разбиране на нуждите и очакванията на клиента
- Идентифициране на изискванията
- Анализ на изискванията
- Дефиниране на изискванията
- Изчистване и уточняване на изискванията
- Приоритизиране на изискванията
- Разпределяне и алокиране на изискванията
- Проследяване на изискванията
- Управление на изискванията
- Тестване и верификация на изискванията
- Валидиране на изискванията

## Документиране на изискванията към софтуера

- eXtreme Programming (XP) – User Stories
- UML – Use Cases (случаи на употреба)
  - описание на поведението на системата като отговори на заявки генерирани извън нея
  - Use Case техниката се използва за описание на изискванията към поведението на системата чрез детайлизиране на поведението под формата на отделни последователности (нишки) от събития, за описване на функционалните изисквания към системата.



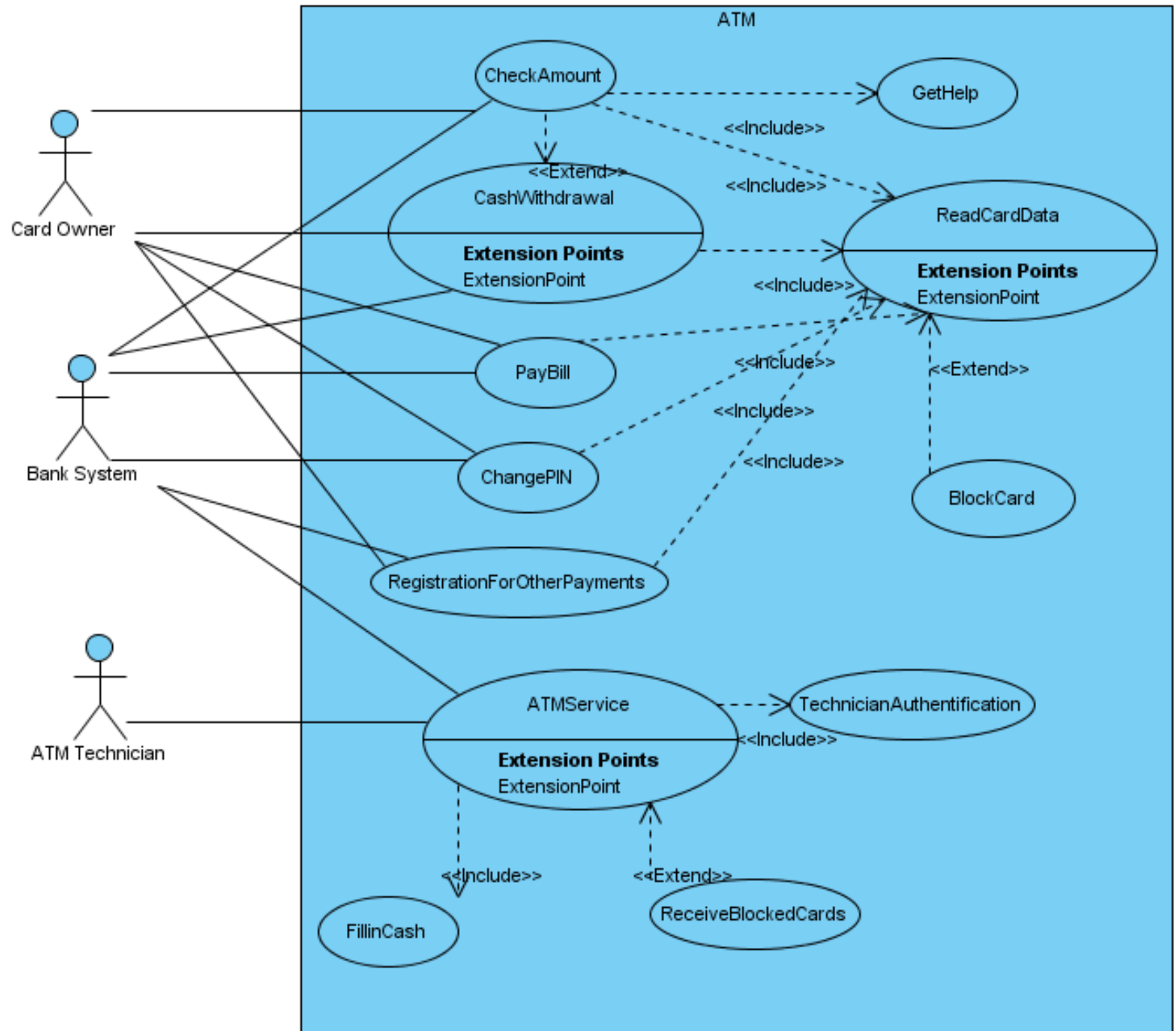
## Пример ...

<b>Name</b>	<u>CashWithdrawal</u>	
<b>Super Use Case</b>		
<b>Primary Actor</b>	Card Owner	
<b>Level</b>	sea-level	
<b>Secondary Actor(s)</b>	Bank system	
<b>Brief Description</b>	<u>CardOwner</u> identifies her/himself to the <i>ATM System</i> . After that he/she chooses the withdrawal operation and the amount of money to be withdrawn. After confirmation he/she receives card, money amount, and receipt.	
<b>Preconditions</b>	<i>ATM</i> is operational and in <i>Ready State</i> and there is Cash Available	
<b>Flow of Events</b>		<b>Actor Input</b>
	1	<i>CO</i> inserts <i>Card</i>
	2	
	3	
	4	<i>CO</i> chooses cash withdrawal operation
	5	
	6	<i>CO</i> choose amount
	7	
	8	
	9	<i>CO</i> enters PIN
	10	
	11	<i>Bank</i> returns transaction acknowledgement
	12	
	13	<i>CO</i> takes <i>Card</i>
	14	
	15	
	16	<i>Bank</i> returns <i>account balance</i>
	17	<i>CO</i> takes cash
	18	
	19	<i>CO</i> takes receipt
	20	
		<b>System Response</b>
		Read <i>Card</i> data using <i>Read Card Data UC</i>
		Present operation <u>choices</u>
		Present amount <u>choices</u>
		Assure cash amount available in ATM
		Ask for PIN
		Send Card data, amount, PIN to Bank
		Return <i>Card</i>
		Dispense cash
		Send confirmation to Bank
		Print and dispense receipt
		Return to Ready State
<b>Post-conditions</b>	<i>ATM</i> system is in <i>Ready State</i> , and cash and Card slots are empty. Or in case the card was invalid or too many attempts for entering PIN the Card is arrested.	
<b>Minimal Guarantee</b>	<i>ATM</i> system is in <i>Ready State</i>	



## Диаграма на потребителските случаи в UML

- Основни елементи
  - Актьори (Actors) – моделират типове роли, изпълнявани от същности (entities), които взаимодействат със субекта (напр. чрез обмяна на сигнали или данни, но които са *външни* за субекта)
  - Субект (Subject) – системата, която се моделира
  - Случаи на употреба (Use Cases)
- Типове връзки между потребителски случаи
  - <<include>>
  - <<extend>>
  - генерализация/специализация



## Видове потребителски случаи

- Според предназначението:
  - бизнес потребителски случаи (business use cases)
  - системни потребителски случаи (system use cases)
- Според нивото на детайлност [Alistair Cockburn]:
  - Brief use case
  - Causal use case
  - Fully dressed use case

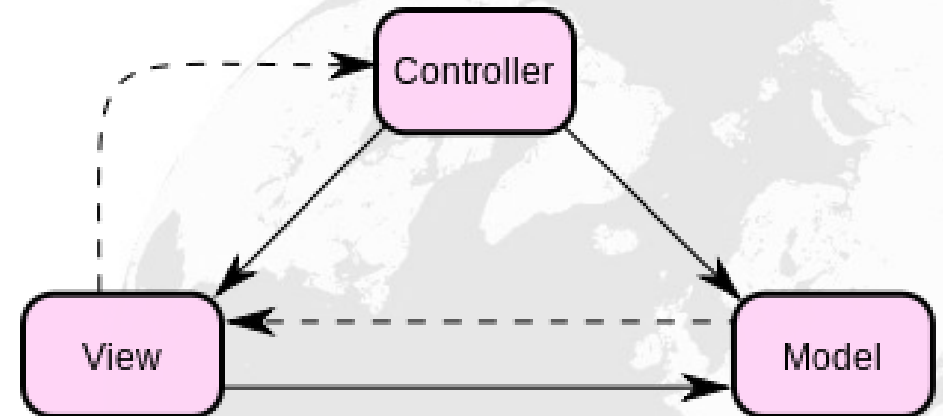
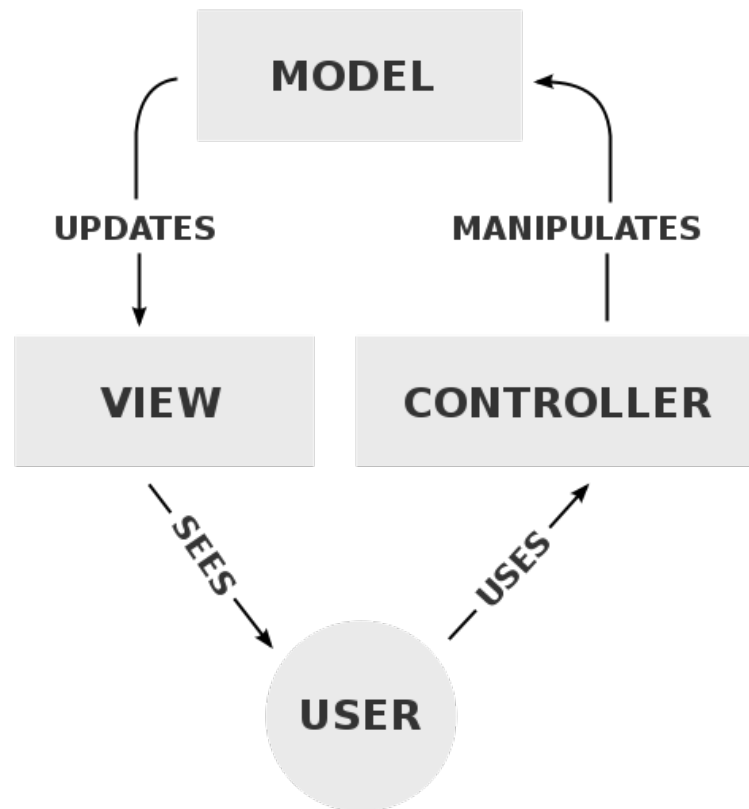
## Шаблони за описание [Wikipedia]

- Име на потреб. случай
- Версия
- Цел
- Кратко описание
- Актьори
- Пред-условия (Preconditions)
- Тригери
- Основен поток на събития
- Алтернативни потоци на събития
- Потоци на събития за обработка на грешки и изключения
- След-условия (Postconditions)
- Бизнес правила
- Забележки
- Автор и дата

## Качество на описание на потребителските случаи

- Да описваме какво трябва да прави системата за да постигне актьорът своята конкретна цел
- Да не се използва език специфичен за имплементацията
- Да се подбере подходящото ниво на детайлност на описанието според етапа от жизнения цикъл и избрания процес за разработка на продукта
- Потребителският случай не трябва да включва детайли относно потребителския интерфейс – описанието да бъде на интенционално ниво.

# Model-View-Controller MVC архитектура



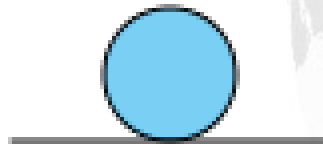
## Стереотипи на анализ класове

Анализ класовете се използват при скицирането и анализа на архитектурата на системата – те представят по-скоро различни роли и отговорности, отколкото конкретни класове, които да бъдат реализирани и са независими от имплементационните технологии:

- <<control>> - бизнес логика
- <<entity>> - модел на данните
- <<boundary>> - потреб. или системен интерфейс



**Контролиращ клас**

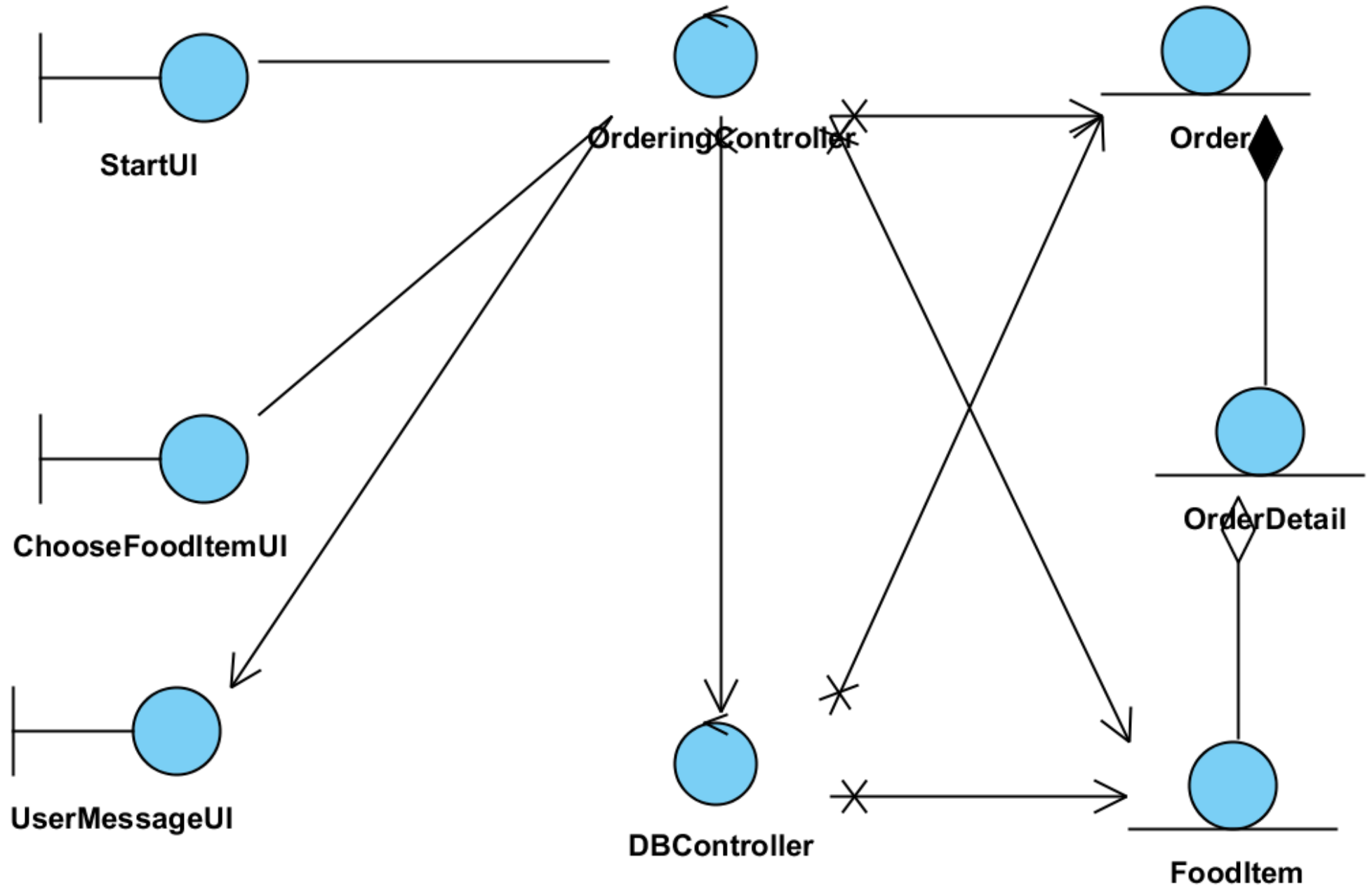


**Клас единица**



**Граничен клас**





## Учебни проекти

- Цел – да се специфицира и реализира бизнес софтуерна система с MVC архитектура и текстови потребителски интерфейс (текстови менюта) от команден ред.
- Задачи:
  - Да се оформи задание описващо накратко основната функционалност на приложението под формата на сценарии за всеки актьор
  - Да се реализира, тества и демонстрира системата с предварително въведени данни

## Оценяване на учебния проект

№	Име на задание	% от оценката
1.	Кратко описание на проекта (Project Summary - example template: ) - необходимо е да бъде дискутирано с преподавателя по време на занятия, записано в списък задания в Google docs: <a href="https://docs.google.com/spreadsheets/d/1V3nrgZk4xDxXc8AUbePvOduY_vvqPHP7Ij40Xc1H5Yw/edit?usp=sharing">https://docs.google.com/spreadsheets/d/1V3nrgZk4xDxXc8AUbePvOduY_vvqPHP7Ij40Xc1H5Yw/edit?usp=sharing</a> и одобрено от преподавателя (колонка "статус").	10%
2.	Реализация на проекта на езика Java	80%
3.	Представяне и защита на проекта	10%

Благодаря Ви за вниманието!

Въпроси?