**МИНИСЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**отделение среднего профессионального образования**

**«Казанский национальный исследовательский технический университет**

**им. А.Н. Туполева-КАИ»**

**(КНИТУ-КАИ)**

**Колледж информационных технологий**

**Реферат**

На тему:

«UML - диаграммы классов»

Пример на основе АИС учета успеваемости студентов

Выполнили:

студенты гр. 4335

Алякимов Б.Ф.

Лебедев Д.М.

Проверил:

Александров А.Ю.

Оценка\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_

Казань, 2017 г.

**Содержание**

[Аннотация 3](#_Toc499592171)

1.[Введение 4](#_Toc499592172)

[2. Краткая характеристика предметной области 6](#_Toc499592173)

[1.1 Общая характеристика 6](#_Toc499592174)

[1.2 Актуальность разрабатываемой подсистемы 6](#_Toc499592175)

[1.3 Формулировка задач проектирования 6](#_Toc499592176)

3. Диаграммы классов

# **Аннотация**

Данный реферат посвящен теме использования UML–диаграмм классов.

Диаграмма классов (англ. Static Structure diagram) — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Диаграмма классов UML позволяет обозначать отношения между классами и их экземплярами. Для чего они нужны? Они нужны, например, для моделирования прикладной области. Но как отношения отражаются в программном коде? Данное небольшое исследование пытается ответить на этот вопрос — показать эти отношения в коде.

1. **Введение**

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

Унифицированный язык моделирования (UML) является стандартным инструментом для создания «чертежей» программного обеспечения. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.

UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени. Это очень выразительный язык, позволяющий рассмотреть систему со всех точек зрения, имеющих отношение к ее разработке и последующему развертыванию. Несмотря на обилие выразительных возможностей, этот язык прост для понимания и использования. Изучение UML удобнее всего начать с его концептуальной модели, которая включает в себя три основных элемента: базовые строительные блоки, правила, определяющие, как эти блоки могут сочетаться между собой, и некоторые общие механизмы языка.

Несмотря на свои достоинства, UML - это всего лишь язык; он является одной из составляющих процесса разработки программного обеспечения, и не более того. Хотя UML не зависит от моделируемой реальности, лучше всего применять его, когда процесс моделирования основан на рассмотрении прецедентов использования, является итеративным и пошаговым, а сама система имеет четко выраженную архитектуру.

UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (англ. generalization), агрегация (англ. aggregation) и поведение) и больше сконцентрироваться на проектировании и архитектуре.

На рисунке 1.1 представлены виды диаграмм, использующиеся в UML (для исключения неоднозначности приведены также обозначения на английском языке):

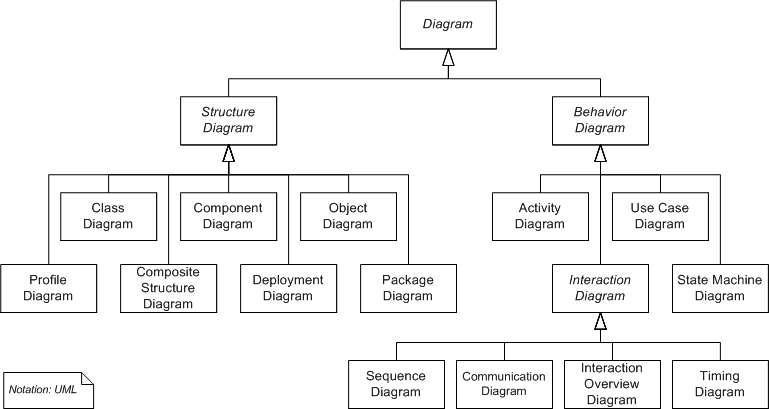


Рис. 1.1

В данном курсовом проекте разработана объектно-ориентированная модель и создана диаграмма классов информационной подсистемы для учета успеваемости студентов факультета.

**2. Краткая характеристика предметной области**

**2.1 Общая характеристика**

Разрабатываемая информационная подсистема предназначена для использования в рамках информационной подсистемы для деканата факультета учебного заведения (учет успеваемости студентов).

**2.2 Актуальность разрабатываемой подсистемы**

В современных условиях рыночно экономики большое значение имеет качество образования. Это приводит к тому, что все больше и больше молодых людей, понимая это, поступают в государственные высшие учебные заведения крупных городов. Такие образовательные учреждения, как правило, имеют статистику каждого факультета, у которой отображается информация об успеваемости тысяч студентов. Так как в «Университете» насчитывается около десятка факультетов то подсчет всех критериев очень труден и требует внедрение и использование электронных баз данных для учета успеваемости студентов.

**2.3 Формулировка задач проектирования**

Для эффективного функционирования системы подсчета успеваемости студентов, информационная подсистема учета студентов должна выполнять ряд задач.

Деканат, в роли которого может выступать секретарь деканата или другое уполномоченное лицо осуществляет оперативный учет студентов. В его обязанности входит:

- добавление информации о студенте в базу данных при поступление его на факультет. В качестве такой информации могут выступать фамилия, имя, отчество студента, дата рождения, факультет, курс и группу;

- периодическое занесение оперативной информации обо всех видах ведомостей (например, экзаменационная, итоговая, аттестационная, промежуточная и т.д.).

Этот небольшой список задач позволяет решить основные вопросы, связанные с учетом успеваемости студентов.

1. **Диаграммы классов**

Class diagram (диаграммы классов) позволяет создавать логическое представление системы, на основе которого создается исходный код описанных классов. На диаграммах классов отображаются некоторые классы и пакеты системы. Это статические картины фрагментов системы и связей между ними.

**Классификация**

Существует два вида диаграмм классов:

* Статический вид диаграммы рассматривает логические взаимосвязи классов между собой;
* Аналитический вид диаграммы рассматривает общий вид и взаимосвязи классов, входящих в систему.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

* концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
* точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
* точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).

**Элементы диаграммы**

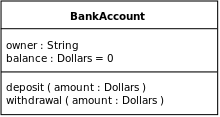
[](https://commons.wikimedia.org/wiki/File:BankAccount1.svg?uselang=ru)

Рис.2.1 Класс с двумя методами и полями.

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента(Рис.2.1):

* В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом.
* Посередине располагаются поля (атрибуты) класса. Они выровнены по левому краю и начинаются с маленькой буквы.
* Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы.