

Федеральное государственное образовательное бюджетное учреждение  
высшего образования

**«Финансовый университет при Правительстве Российской  
Федерации»**

**(Финансовый университет)**

Факультет информационных технологий и анализа больших данных

**Департамент анализа данных и машинного обучения**

**Контрольная работа №2**

**по дисциплине «Проектирование информационных систем»**

**на тему:**

**«Визуальное моделирование системы учета оценок студентов в StarUML»**

**Выполнили:**

Студентки гр. ПИ19-1

Воронина Ксения Максимовна

Коновалова Анастасия Сергеевна

**Преподаватель:**

Медведев А. В.,

доцент, кандидат экономических наук

**Москва, 2021 г.**

## **Оглавление**

<b>Введение.....</b>	<b>3</b>
<b>Диаграмма вариантов использования .....</b>	<b>4</b>
<b>Потоки событий .....</b>	<b>6</b>
<b>Диаграмма действий .....</b>	<b>10</b>
<b>Диаграмма классов.....</b>	<b>11</b>
<b>Заключение .....</b>	<b>13</b>

## Введение

Компьютерные технологии стали неотъемлемой частью жизнедеятельности современного человека. Компьютерная техника используется человеком в повседневной жизни для учебы, отдыха и для осуществления профессиональной деятельности.

Для более полного использования вычислительных мощностей современного компьютера создаются разнообразные программные продукты и информационные системы, которые предназначены для решения определенного круга задач.

Процесс создания информационной системы проходит несколько этапов: от постановки задачи и создания проекта, до непосредственной разработки и внедрения её в производственный процесс. В данной работе будет рассмотрен этап проектирования информационной системы на примере информационной системы «Система учета оценок учеников».

То есть объектом исследования является проектирование информационных систем, а предметом исследования – информационная система «Система учета оценок учеников».

Основная цель – создание проекта информационной системы «Система учета оценок учеников» с помощью языка UML.

Рассматривается система учета оценок, целью деятельности которой является информирование и ведения учета. Учет ведется по названиям дисциплин и системе оценок.

Работники деканата преподаватели отправляют данные студентов для составления системы оценок, где используются курс, форма обучения направление и успеваемость студента по дисциплинам.

Для достижения поставленной цели необходимо решить следующие задачи:

- сформулировать понятие объектно-ориентированного подхода и выявить основные его элементы;
- выделить достоинства и недостатки объектно-ориентированного подхода;
- разработать диаграммы для ИС «Система учета оценок учеников»: диаграмма вариантов использования, диаграммы деятельности, диаграмма классов.

*Унифицированный язык моделирования (Unified Modeling Language, UML)* является графическим языком для визуализации, специфицирования, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению.

*Основные диаграммы UML:* вариантов использования (use case diagram); классов (class diagram); кооперации (collaboration diagram); последовательности (sequence diagram); состояний (statechart diagram); деятельности (activity diagram); компонентов (component diagram);

развертывания (deployment diagram). Построения этих диаграмм достаточно для полного моделирования системы.

## Диаграмма вариантов использования

Построение объектно-ориентированной модели на языке UML чаще всего начинают с составления диаграммы прецедентов (или вариантов использования).

Диаграммы вариантов использования применяются для моделирования вида системы с точки зрения внешнего наблюдателя. На диаграмме прецедентов графически показана совокупность прецедентов и Субъектов, а также отношения между ними.

Поведение системы (т.е. функциональность, которую она обеспечивает) описывают с помощью функциональной модели, которая отображает системные прецеденты (use cases, случаи использования), системное окружение (действующих лиц, актеров, actors) и связи между ними (use cases diagrams).

*Диаграмма вариантов использования (диаграмма прецедентов, use case diagram)* — это диаграмма, на которой изображаются отношения между актерами и вариантами использования.

*Актером (действующее лицо, actor)* называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определяет разработчик. На диаграммах вариантов использования актер изображается в виде человечка, под которым записывается его имя.

*Вариант использования (прецедент, use case)* — описание множества последовательности действий (включая варианты), выполняемых системой для того, чтобы актер мог получить определенный результат.

Каждый вариант использования должен быть независимым в том смысле, что если он всегда выполняется совместно с другим вариантом использования, то, скорее всего, это один прецедент, а не два, либо они связаны отношением включения или расширения.

Прецедент описывает, что делает система, но никак не уточняет, как она это делает. Заметим, что диаграмма прецедентов не отображает последовательность, в которой будут выполняться варианты использования. На диаграмме прецедент изображается в виде эллипса. Имя прецедента может состоять из нескольких слов и знаков препинания (за исключением 16 двоеточия), как правило, имя выбирают в виде словосочетания или глагольного выражения.

Моделирование прецедентов и актеров помогает нам лучше понять требования, предъявляемые к системе, и согласовать их с заказчиком с помощью демонстрации и обсуждения диаграммы прецедентов.

Прецеденты и актеры – это отражение требований к системе, они показывают, кто и для чего будет использовать будущую систему.

*Отношение (relationship)* — это семантическая связь между отдельными элементами модели.

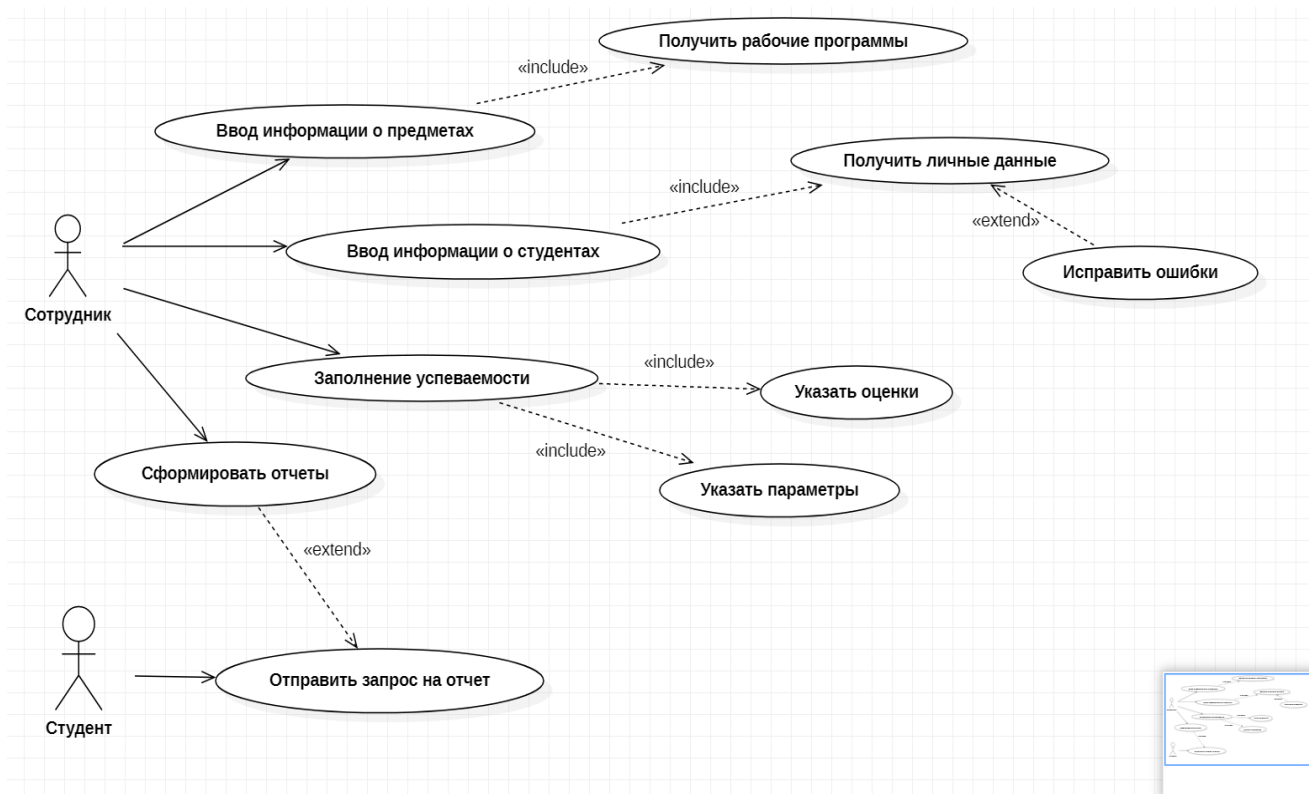
В основном на диаграммах прецедентов используются следующие *типы отношений*: ассоциации (association relationship); включения (include relationship); расширения (extend relationship); обобщения (generalization relationship).

*Ассоциация* – это структурное отношение, показывающее, что объект неким образом связан с другим объектом.

*Включение (include)* говорит о том, что исходный прецедент явным образом включает в себя поведение целевого. Изображается включение в виде пунктирной стрелки с надписью <include>, которая направлена от базового элемента к используемому. (Например, заказ товара включает в себя оплату товара, без нее заказ невозможен. Стрелка идет от главного действия - заказ товара - к включенному действию - оплата товара.)

*Расширение (extend)* показывает, что целевой прецедент расширяет поведение исходного. Используемый прецедент выполняется не всегда вместе с базовым, а только при выполнении дополнительных условий, таким образом, расширяя функциональность базового элемента. Изображается расширение пунктирной стрелкой с надписью <extend>, направленной от используемого варианта использования к базовому.

*Обобщение (Generalization)* – это отношение между общей сущностью и ее конкретным воплощением. На диаграммах обобщение обозначается стрелкой с не закрашенным треугольником на конце, направленной от частного элемента к общему.



## Потоки событий

*Поток событий* – это определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста. Потоки событий – это текстовые описания пошагового выполнения прецедентов, они понятны не только разработчику, но и стороннему читателю.

*Основной (главный) поток* описывает наилучший сценарий либо наиболее используемый путь исполнения прецедента. *Альтернативный поток* специфицирует отклонения от основного потока, которые не рассматриваются как ошибочные. *Поток ошибок* рассматривается как отклонение от альтернативного или основного, которое порождает условия формирования ошибки.

### Потоки событий “Ввод информации о предметах”

#### Основной поток событий

1. Прецедент начинается с выбора сотрудником режима ввода информации о предмете;
2. Система открывает базу данных контрольных мероприятий по предметам;

3. *E1 "Error 404"*;
4. Сотрудник нажимает кнопку "добавить";
5. Запрос рабочих программ;
6. *A1 "Рабочая программа не найдена"*;
7. Сотрудник вводит информацию по датам и времени;
8. Сотрудник нажимает кнопку "записать и закрыть";
9. Программа успешно добавляет новую запись;
10. *A2 "Ошибка в заполнении данных"*.

### **Альтернативные потоки событий**

#### *A1 "Рабочая программа не найдена"*

1. Система не смогла найти нужную рабочую программу;
2. Сотрудник исправляет параметры поиска;
3. Переход в п.5 основного потока.

#### *A2 "Ошибка в заполнении данных"*

1. Система выдает ошибку заполнения данных;
2. Сотрудник исправляется введенные данные;
3. Переход в п.9 основного потока.

### **Потоки ошибок**

#### *E1 "Error 404"*

1. Ошибка на сервере, информация по предметам недоступна;
2. Сотрудник завершает работу на неопределенное время.

### **Потоки событий "Ввод информации о студентах"**

#### **Основной поток событий**

1. Прецедент начинается с выбора сотрудником режима ввода информации о студентах;
2. Система открывает базу данных;
3. *E1 "Error 404"*;
4. Сотрудник получает личные данные студента;
5. Сотрудник проверяет корректность данных;
6. *A1 "Данные некорректны"*;
7. Сотрудник нажимает кнопку "добавить";
8. Сотрудник вводит полученную информацию о студенте;
9. Сотрудник нажимает кнопку "записать и закрыть";
10. Программа успешно добавляет новую запись;
11. *A2 "Ошибка в заполнении данных"*.

## **Альтернативные потоки событий**

### *A1 “Данные некорректны”*

1. Сотрудник обнаружил ошибку в личных данных;
2. Сотрудник получает исправленные данные;
3. Переход в п.7 основного потока.

### *A2 “Ошибка в заполнении данных”*

1. Система выдает ошибку заполнения данных;
2. Сотрудник исправляет введенные данные;
3. Переход в п.10 основного потока.

## **Потоки ошибок**

### *E1 “Error 404”*

1. Ошибка на сервере, информация по предметам недоступна;
2. Сотрудник завершает работу на неопределенное время.

## **Потоки событий “Заполнение успеваемости”**

### **Основной поток событий**

1. Прецедент начинается с выбора сотрудником режима ввода информации о результатах контрольных мероприятий;
2. Система открывает базу данных;
3. *E1 “Error 404”*;
4. Сотрудник получает контрольные ведомости от преподавателей;
5. Сотрудник проверяет корректность данных;
6. *A1 “Данные некорректны”*;
7. Сотрудник нажимает кнопку “добавить”;
8. Сотрудник выбирает студента в системе;
9. *A2 “Студент не найден”*;
10. Сотрудник выбирает предмет в системе;
11. *A3 “Предмет не найден”*;
12. Сотрудник выставляет результат контрольного мероприятия;
13. Сотрудник нажимает кнопку “записать и закрыть”;
14. Программа успешно добавляет новую запись;
15. *A2 “Ошибка в заполнении данных”*.

## **Альтернативные потоки событий**



#### *A1 “Данные некорректны”*

1. Сотрудник обнаружил ошибку в данных;
2. Сотрудник получает исправленные данные;
3. Переход в п.7 основного потока.

#### *A2 “Студент не найден”*

1. Сотрудник переходит в прецеденту “Ввод информации о студенте”;
2. Переход в п.8 основного потока.

#### *A3 “Предмет не найден”*

1. Сотрудник переходит в прецеденту “Ввод информации о предмете”;
2. Переход в п.10 основного потока.

#### *A4 “Ошибка в заполнении данных”*

1. Система выдает ошибку заполнения данных;
2. Сотрудник исправляется введенные данные;
3. Переход в п.12 основного потока.

### **Потоки ошибок**

#### *E1 “Error 404”*

1. Ошибка на сервере, информация по предметам недоступна;
2. Сотрудник завершает работу на неопределенное время.

### **Поток событий “Сформировать отчет”**

#### **Основной поток событий**

1. Прецедент начинается с получения сотрудником запроса на формирование отчета;
2. Сотрудник открывает страницу формирования отчетов;
3. *E1 “Error 404”*;
4. Сотрудник выбирает нужную форму отчета;
5. Сотрудник вводит данные для отчета;
6. *A1 “Данные некорректны”*
7. Сотрудник нажимает кнопку “сформировать отчет”;
8. Прецедент завершен;
9. *A2 “Печать отчета”*

#### **Альтернативные потоки**

### *A1 “Данные некорректны”*

1. Сотрудник или система обнаружил ошибку в данных;
2. Сотрудник получает исправленные данные;
3. Переход в п.7 основного потока.

### *A2 “Печать отчета”*

1. Сотрудник выбирает опцию “печать”;
2. Печать отчета;
3. Переход в п.8 основного потока.

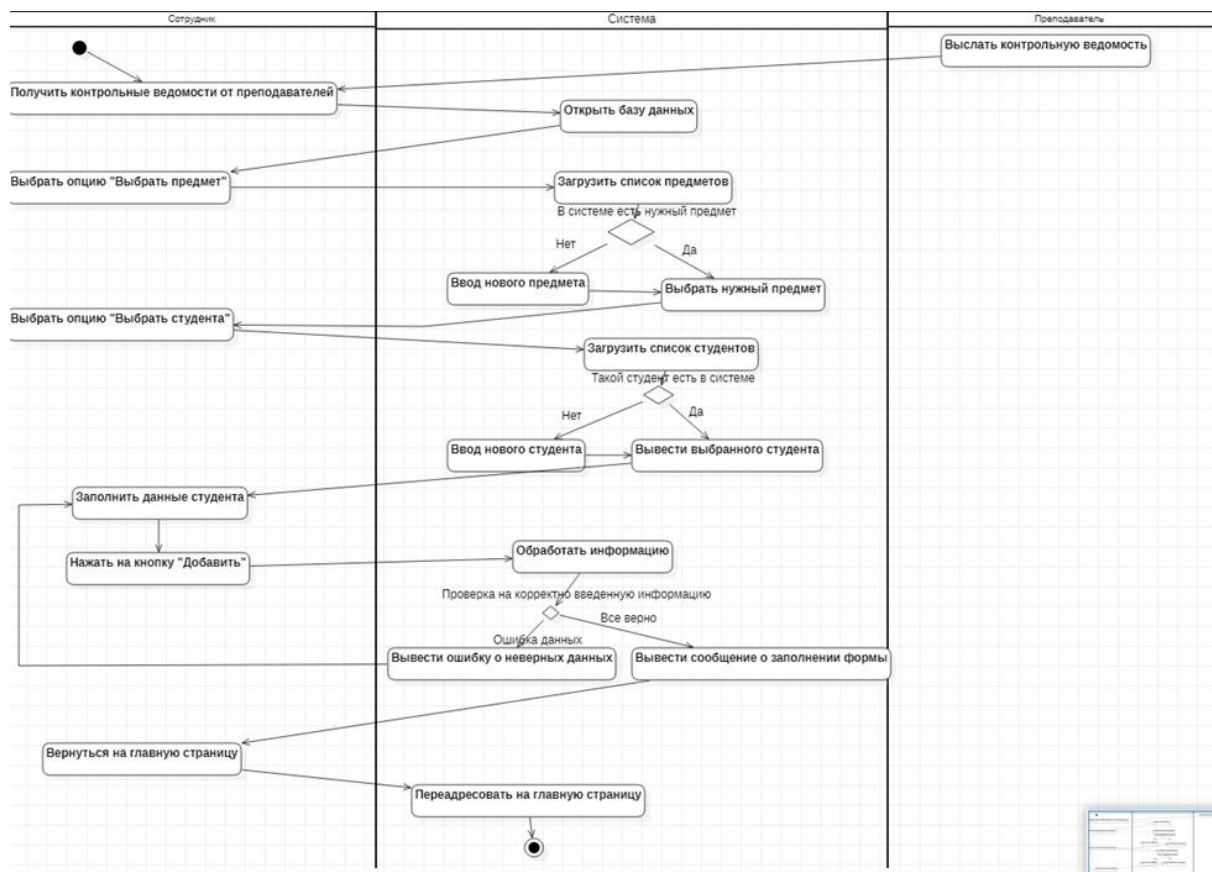
### *E1 “Error 404”*

1. Ошибка на сервере, информация по предметам недоступна;
2. Сотрудник завершает работу на неопределенное время.

## **Диаграмма действий**

Каждый вариант использования, представленный на диаграмме прецедентов, нуждается в уточнении. Это объяснение дают диаграммы деятельности, которые рассматривают каждый вариант использования в виде последовательности действий. Дополнительным и очень важным элементом диаграммы являются ветвления, которые позволяют описать альтернативные варианты поведения пользователя.

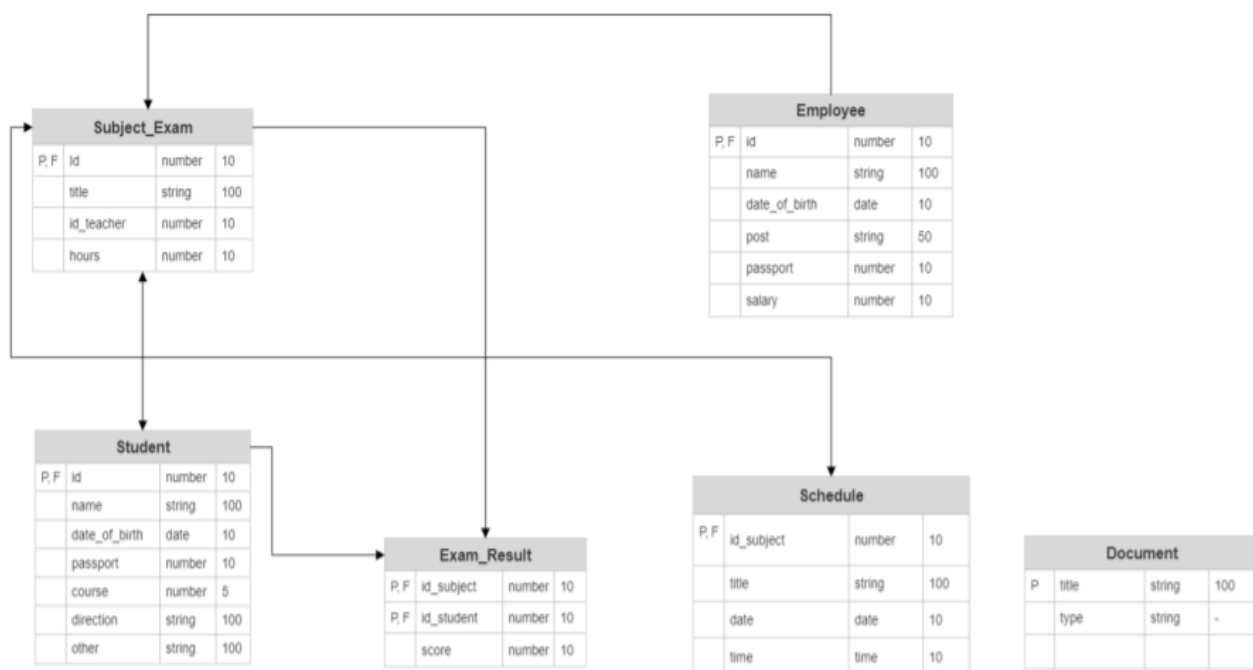
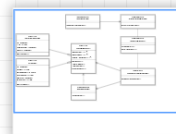
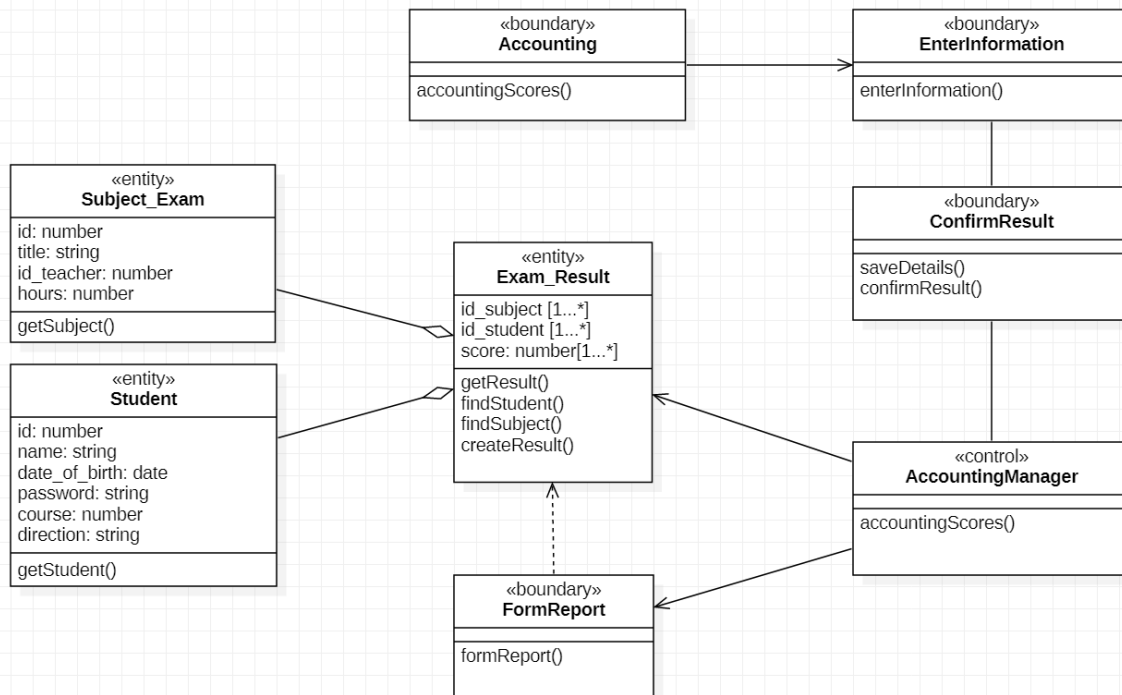
Старт выполнения прецедента обозначается заполненным кружком, а конец – кружком в окружности. Старт у прецедента может быть только один на всю диаграмму деятельности, тогда как точек окончания может быть сколько угодно (но не менее одной). Сами действия обозначаются скругленными прямоугольниками, а ветвления – ромбами.



## Диаграмма классов

Класс – это основной составной элемент информационной системы. Данное понятие является одним из основополагающих в объектно-ориентированных языках программирования. Соответственно, между классами в диаграммах UML и классами в программном коде есть соответствие, поэтому большинство средств построения диаграмм UML позволяют генерировать программный код на выбранном языке программирования на основе диаграммы классов.

Каждый класс обязательно имеет уникальное название, а также атрибуты и методы. Атрибутами называются свойства класса, а методами – действия, которые может выполнять объект выбранного класса, чаще всего методы класса предназначены для изменения его свойств.



## Заключение

В ходе работы были исследованы особенности объектно-ориентированного подхода к проектированию информационных систем.

Были исследованы основные элементы, которые составляют объектно-ориентированную модель, а также особенности их выделения и взаимодействия друг с другом. Для реализации объектно-ориентированного подхода был разработан специальный язык – UML, с помощью которого и была спроектирована информационная система «Система учета оценок учеников».

Согласно данному языку, объектная модель состоит из взаимосвязанных диаграмм разного типа, каждая из которых показывает информационную систему с определенной стороны (либо отображает только её часть).

Для проекта информационной системы были разработаны следующие диаграммы:

- диаграмма вариантов использования;
- диаграммы деятельности;
- диаграмма классов.

Таким образом, все задачи проекта выполнены, а значит и основная цель достигнута.