

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курский государственный университет»

кафедра программного обеспечения и администрирования
информационных систем

Отчёт
по контрольной работе № 3

«Основы объектно-ориентированного анализа и проектирования»

по дисциплине

«Объектно-ориентированные языки и системы»

Выполнил:

студент группы 313

Мусонда Салиму

Проверил:

к.т.н., доцент

кафедры ПОиАИС

Макаров К.С.

Курск
2020

Цель работы: разработка модели вариантов использования и диаграммы классов для решения задач проектирования программной системы.

Теоретические сведения:

Отдельные варианты использования могут применяться как для спецификации требований к проектируемой системе, так и для документирования процесса поведения имеющейся системы. Кроме этого, варианты использования неявно специфицируют требования, определяющие особенности взаимодействия пользователей с системой, которые специфицируют возможность корректной работы с предоставляемыми данной системой сервисами.

Требование (requirement) – желательное свойство, характеристика или условие, которым должна удовлетворять система в процессе своей эксплуатации.

Применительно к программным системам предложена следующая классификация требований, которая получила название модели FURPS+, что соответствует первым буквам соответствующих категорий требований на английском языке:

- функциональные требования (Functionality);
- требования удобства использования (Usability);
- требования надежности (Reliability);
- требования производительности (Performance);
- требования возможности сопровождения (Supportability).

При этом символом «+» обозначены дополнительные условия, к которым относятся:

- проектные ограничения;
- требования управления системой;
- требования к графическому интерфейсу пользователя;
- физические требования;
- юридические требования.

Центральное место среди указанных требований занимают функциональные, которые специфицируют особенности реализации отдельных бизнес-

процессов моделируемой системы. В контексте моделей языка UML именно функциональные требования должны служить исходной информацией для построения диаграмм вариантов использования. Однако графических средств языка UML на практике оказывается недостаточно для спецификации функциональных требований.

Следует отметить, что одним из требований языка UML является самодостаточность диаграмм для представления информации о моделях проектируемых систем. Однако большинство разработчиков и экспертов согласны с тем, что изобразительных средств языка UML явно не хватает для того, чтобы учесть на диаграммах вариантов использования особенности функционального поведения сложной системы. С этой целью рекомендуется дополнять этот тип диаграмм текстовыми сценариями, которые уточняют или детализируют последовательность действий, совершаемых системой при выполнении ее вариантов использования.

Сценарий (scenario) – определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста.

В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования.

Задание:

Разработать модель вариантов использования и диаграмму класса для программной системы в соответствии с вариантом задания.

Для модели вариантов использования предусмотреть минимум 5 диаграмм вариантов использования и соответствующих сценарием. Отразить связи между вариантами использования.

Вариант Школа

Требуется разработать программную систему, предназначенную для завуча школы. Она должна обеспечивать хранение сведений о каждом учителе, о предметах, которые он преподает, номере закрепленного за ним кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какую оценку имеет в текущей четверти по каждому предмету. Завуч должен иметь возможность добавить сведения о новом учителе или ученике, внести в базу данных четвертные оценки учеников каждого класса по каждому предмету, удалить данные об уволившемся учителе и отчисленном из школы ученике.

Диаграмма вариантов использования представлена на рисунке 1.

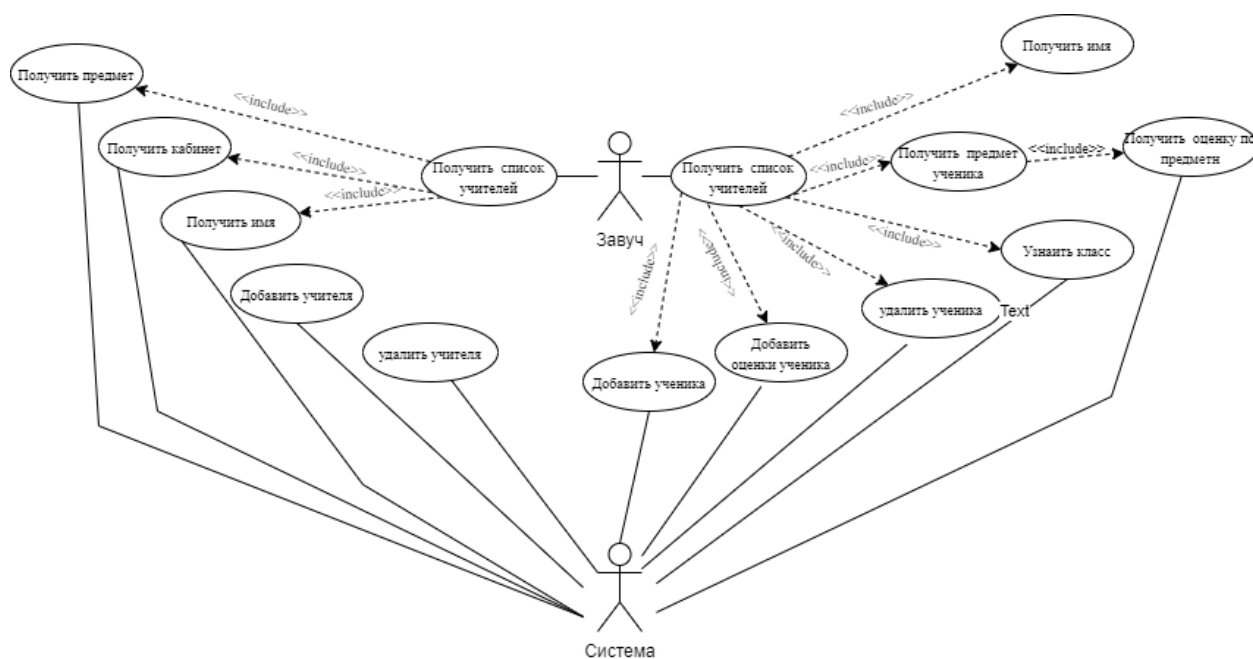


Рисунок 1- Диаграмма вариантов использования

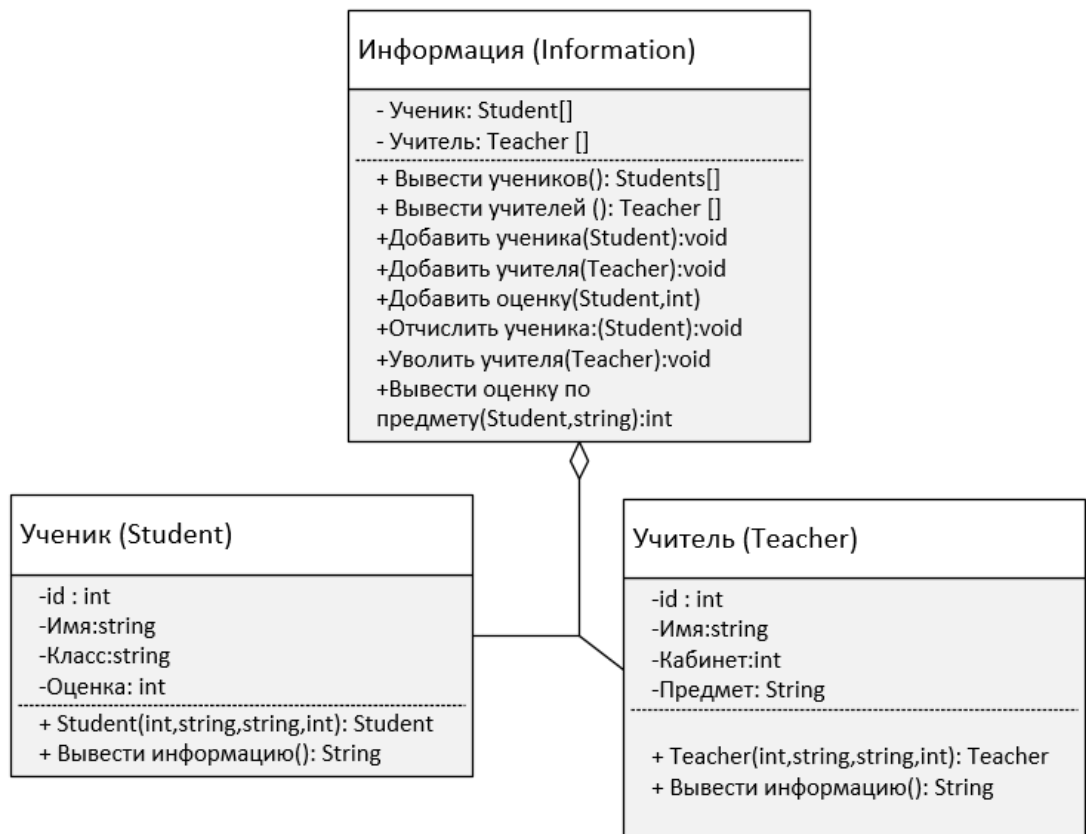


Рисунок 2 – Диаграмма классов

Таблица 1 – Сценарий выполнения варианта использования
«Посмотреть список учителей»

Вариант использования	Посмотреть список учителей
Актеры	Завуч
Цель	Получить информацию о учителях школы
Краткое описание	Завуч запрашивает список учителей
Тип	Базовый
Ссылки на другие варианты использования	Включает в себя ВИ: -Узнать предмет -Узнать закрепленный кабинет -Получить ФИО учителя -Добавить учителя -Удалить учителя
Действия актеров	Отклик системы
1. Завуч подключается к базе данных Исключение №1: Неверные данные	2. Система проверяет корректность введенных данных 3. Допускает завуча до базы данных
4. Завуч выбирает какие данные получить	5. Система предоставляет данные
Исключение №1 Введены неверные данные	
	6. Вывод на экран сообщения об ошибке

Таблица 2 – Сценарий выполнения варианта использования
«Посмотреть список учеников»

Вариант использования	Посмотреть список учеников
Актеры	Завуч
Цель	Получить информацию о учениках школы
Краткое описание	Завуч запрашивает список учеников
Тип	Базовый
Ссылки на другие варианты использования	Включает в себя ВИ: - Узнать класс - Посмотреть список предметов, изучаемых учеником - Узнать оценку по выбранному предмету - Получить ФИО ученика - Добавить ученика - Удалить ученика - Добавить оценку ученика
Действия актеров	Отклик системы
1. Завуч подключается к базе данных учеников Исключение №1: Неверные данные	2. Система проверяет корректность введенных данных 3. Допускает завуча до базы данных
4. Завуч выбирает какие данные получить	5. Система предоставляет данные
Исключение №1 Введены неверные данные	
	6. Вывод на экран сообщения об ошибке

Выводы: я изучил основные принципы и выполнил разработку модели вариантов использования и диаграммы классов для решения задач проектирования программной системы.