**Министерство науки и высшего образования Российской Федерации**

**Университет ИТМО**

**Факультет инфокоммуникационных технологий**

Домашнее задание

на тему

«Разработка программного прототипа по проекту

инфокоммуникационной системы»

по дисциплине:

«Проектирование инфокоммуникационных систем»

Выполнил:

Матвеев Геннадий Андреевич

группа K4110с

Санкт-Петербург

2020

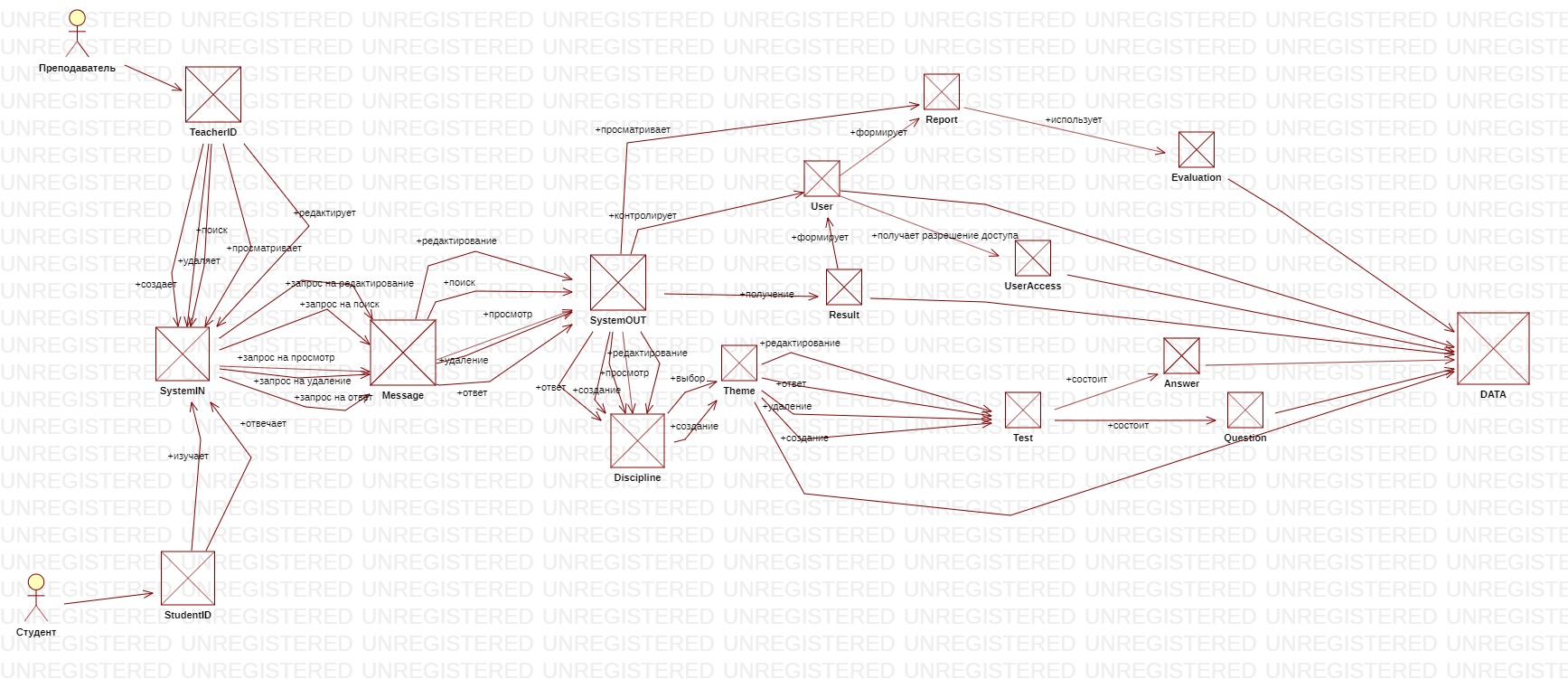
Задание:

Разработать уточненную (после проведения практических занятий и лабораторных работ) диаграмму классов. Реализовать классы как можно подробнее.

Применить шаблоны GoF в тех случаях, где это полезно. Но для обучения и оценивания, чем больше, тем лучше (минимум должно быть реализовано 4 шаблона).

В отчете указать обоснование применения этих шаблонов.

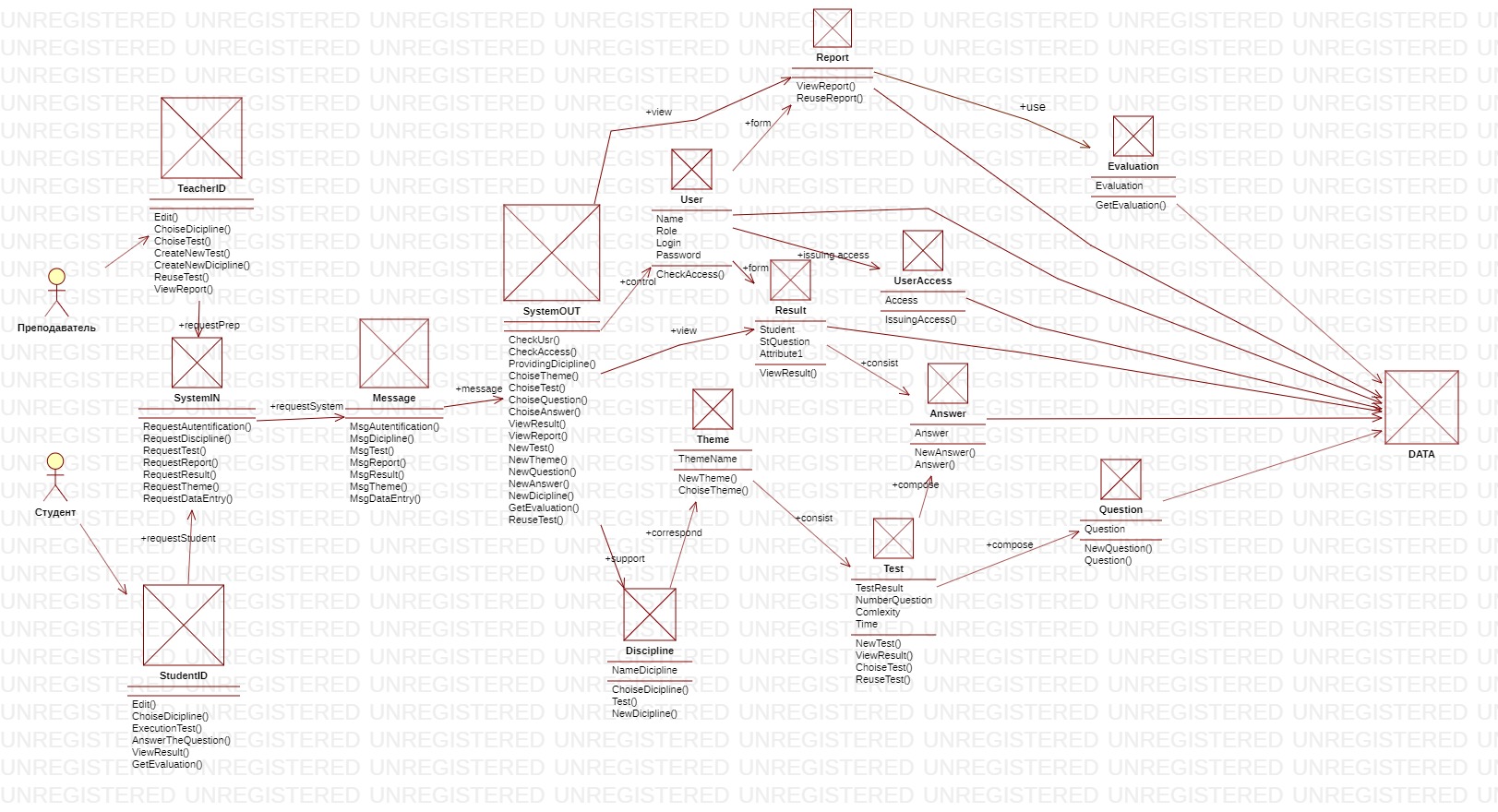
Ниже представлена диаграмма классов и их описание в таблице.



|  |  |  |
| --- | --- | --- |
| № | Наименование класса | Описание |
| *слой представления* | | |
| 1 | PrepID | Граничный класс, отвечающий за отображение формы со списком вопросов, набором ответом и уровня сложности для каждого из них, времени прохождения теста, с возможностью добавления новых вопросов, изменения времени тестирования, уровня сложности и количества вопросов в тесте. |
| 2 | StudentID | Граничный класс, отвечающий за отображение формы со списком вопросов и нескольких вариантов ответов для каждого из вопросов, с возможностью выбора одного варианта ответа, с выводом оставшегося до конца тестирования времени и отображением оценки за результат. |
| 3 | SystemIN | Управляющий класс, методы которого отвечают за управление приложением в целом |
| *слой предметной области* | | |
| 4 | Message | Граничный класс, отвечающий за взаимодействие с классами слоя предметной области |
| 5 | SystemOUT | Управляющий класс, методы которого отвечают за управление приложением системой в целом |
| 6 | Result | Класс хранения, содержащий результат прохождения теста студентом, а именно выбранные им варианты ответов |
| 7 | User | Класс хранения, содержащий список преподавателей и студентов |
| 8 | UserAccess | Класс хранения, содержащий права доступа пользователей системы |
| 9 | Report | Класс хранения, содержащий информацию о проводимых тестах и их результатах |
| 10 | Eval | Класс хранения, содержащий оценки за результаты тестирования |
| 11 | Discipline | Класс хранения, содержащий информацию о конкретной дисциплине |
| 12 | Test | Класс хранения, содержащий информацию о текущем тестировании по конкретной теме, количестве вопросов в тесте, их сложности и времени, которое дано на выполнение данного теста |
| 13 | Answer | Класс хранения, содержащий набор ответов |
| 14 | Question | Класс хранения, содержащий вопросы для тестирования |
| 15 | Theme | Класс хранения, содержащий список тем по дисциплине |
| 16 | Theory | Класс хранения, содержащий учебные материалы лекционных занятий и примеры решения практических заданий |
| *слой источника данных* | | |
| 17 | DATA | Граничный класс, отвечающий за поставку данных из БД для объектов предметной области, сохранение изменений (ввод) в БД. |

Далее была создана уточнённая диаграмма классов, на которой можно увидеть атрибуты и методы классов.

Так, например, у класса слоя предметной области User есть такие атрибуты как Name (имя пользователя), Role (роль пользователя, показывающая какие права у него есть: основные затрагиваемые роли — это преподаватель и студент), Login (логин пользователя), Password (пароль пользователя). У данного класса есть метод CheckAccess (проверка доступа), который проверяет есть ли в базе пользователь с таким логином и паролем и отправляет запрос классу UserAccess на получение доступа. Сам класс User контролирует класс SystemOUT, который отвечает управление системой в целом. Все подобные процессы можно отследить на диаграмме, представленной ниже.



К моему проекту можно применить шаблоны проектирования GoF.

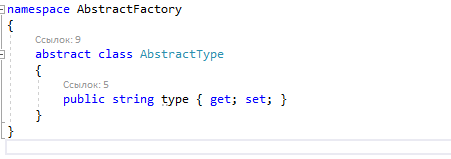
1. Одним из таких шаблонов может быть фабричный метод. Фабричный метод (Factory Method) – порождающий шаблон проектирования, предоставляющий подклассам абстрактный интерфейс (набор методов) для создания экземпляров некоторого класса (объекта-продукта). В момент создания наследники могут самостоятельно принять решение о том, экземпляр какого конкретного класса-продукта создать. Это позволяет использовать в коде программы не специфические классы, а манипулировать абстрактными объектами на более высоком уровне. Таким образом, паттерн Factory Method позволяет базовым абстрактным классам передать ответственность за создание объектов-продуктов своим производным классам.

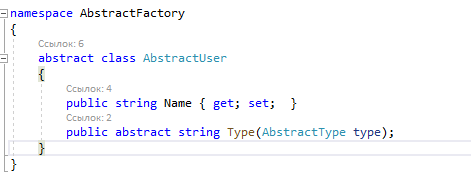
В системе тестирования студентов данный шаблон может применяться следующим образом: классы Question и Answer является наследником класса Test. Данный шаблон делегирует создание объектов наследникам родительского класса.

2. Ещё одним из паттерном, который можно задействовать в работе является Фасад (Facade). Он представляет структурный шаблон проектирования, который позволяет скрыть сложность системы с помощью предоставления упрощенного интерфейса для взаимодействия с ней путем сведения всех возможных внешних вызовов к одному объекту, делегирующему их соответствующим объектам системы. В моём проекте он представлен следующим образом: запросы от преподавателей и студентов сначала поступают в управляющий класс SystemIN, методы которого отвечают за управление приложением в целом, далее с помощью класса Message, отвечающего за взамодействие классов предметной области, информация поступает в управляющий класс SystemOUT, методы которого отвечают за управление приложением системой в целом. Таким образом управление системой произходит за счёт класса SystemOUT, который распределяет запросы пользователей между классами предметной области.

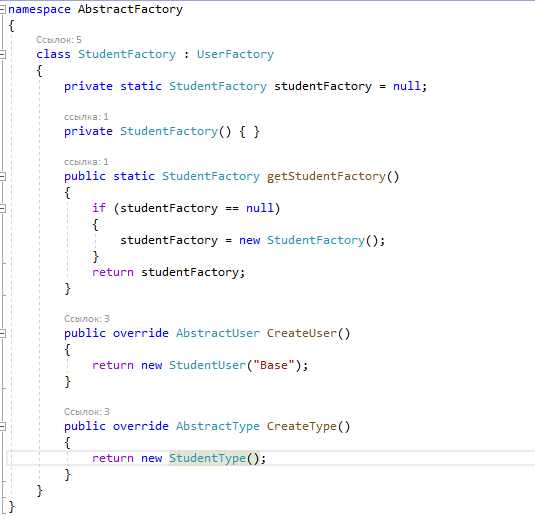
3. Третьим шаблоном, который может быть использован в системе тестирования студентов является Декоратор (Decorator). Декоратор представляет структурный шаблон проектирования, который позволяет динамически подключать к объекту дополнительную функциональность, т.е. расширяет поведение объекта во время выполнения, добавляя или изменяя операции, которые будут осуществляться при обработке запроса. В моём проекте данный шбалон может быть использован для редактирования тестов преподавателем (изменения банка вопросов, сложности вопросов, времени на тест, изменение отчёта для студента). Результаты студентов, прошедших тест должны либо пересматриваться по новым условиям, либо оставаться теми же в зависимости от решения преподавателя.

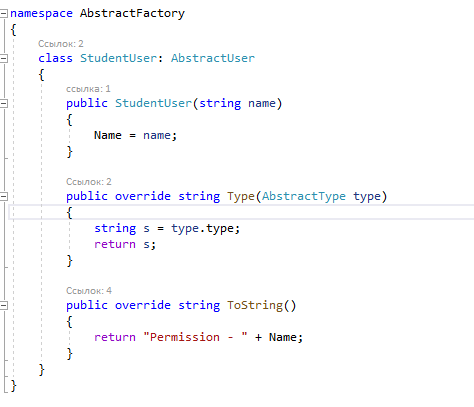
4. Также в работе может быть зайдествован паттерн Абстрактная фабрика (Abstract Factory). Этот порождающий шаблон проектирования предоставляет интерфейс для создания семейств взаимосвязанных объектов с определенными интерфейсами без указания конкретных типов данных объектов, что позволяет разработчику создать интерфейс для объектов, каким-либо образом связанных между собой, причем не требуется указывать конкретные классы, поскольку работать с каждым из них можно будет через этот интерфейс. В системе тестирования студентов таким образом может создан интерфейс под разных пользователей (преподаватель/студент), обладающих разными возможностями в системе.

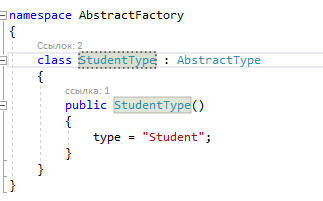


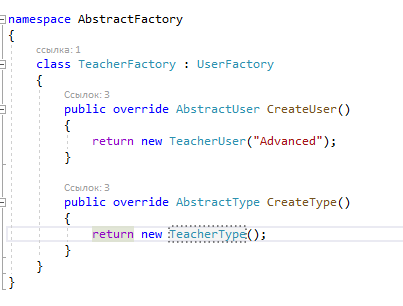


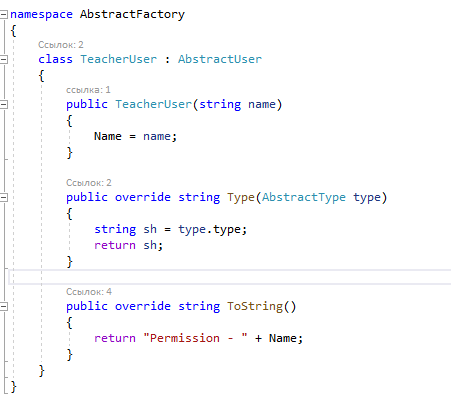


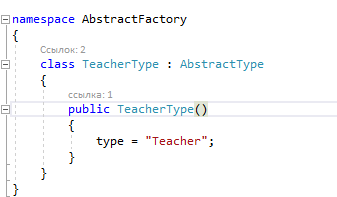


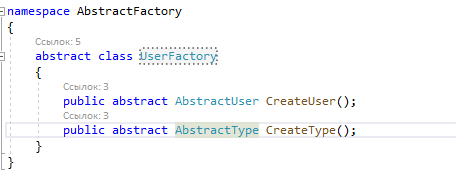


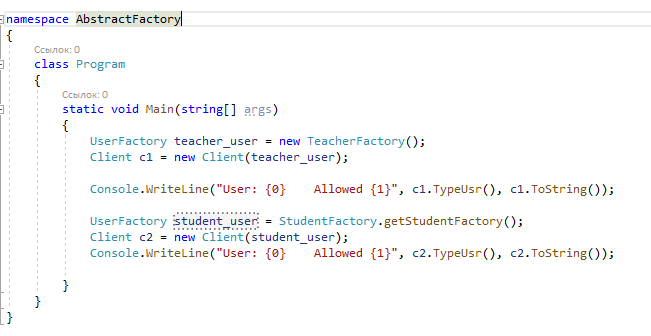












Результат:



Вывод:

В результате выполнения домашнего задания была разработана и объяснена более детальная диаграмма классов, а также описано применение шаблонов GoF (4 описано, один реализован).