Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление: 09.03.02 «Информационные системы и технологии»

Профиль бакалавриата: «Информационные системы и технологии»

**ПРАКТИЧЕСКАЯ РАБОТА**

по дисциплине

**«КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ»**

Тема: **«Проектирование хранилища данных КИС»**

Выполнили:

студенты гр. ЦТУ-20-2б

Никулина Дарья Илларионовна

(Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Лимонова Мария Владимировна

(Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Принял:

ст. преподаватель, Банников Р.Ю.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(должность, ФИО руководителя)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Пермь 2024**

Содержание

[1. Постановка задачи 3](#_Toc161346151)

[1.1. Описание КИС и основного функционала 3](#_Toc161346152)

[1.2. Цель работы и задачи 3](#_Toc161346153)

[2. Диаграмма классов 4](#_Toc161346154)

[2.1. Средство разработки 4](#_Toc161346155)

[2.2. Описание диаграммы классов 4](#_Toc161346156)

[3. Выводы 5](#_Toc161346157)

[3.1. Выводы по проделанной работе 5](#_Toc161346158)

[3.2. Достоинства и недостатки реализованных компонентов 5](#_Toc161346159)

[3.3. Обозначение дальнейших действий 5](#_Toc161346160)

# Постановка задачи

## Описание КИС и основного функционала

Наименование системы: «Модуль корпоративной информационной системы функционирования филиалов онлайн-школы “Exam Masters”».

Модуль корпоративной информационной системы онлайн-школы предназначен для обеспечения пользователям удобного учебного процесса как со стороны учеников, так и со стороны преподавателей и руководства.

Внутри системы пользователи смогут:

1. Добавлять данные в разделы в соответствии с ТЗ;
2. Просматривать статистики;
3. Выгружать отчеты, описанные в ТЗ.

В системе существует разграничение по правам пользователей. Преподавателям доступны следующие функции:

1. Просмотр групп, у которых он ведет занятия;
2. Просмотр каждого отдельного ученика в группе, анализ его статистики;
3. Выставление оценок, корректирование данных в соответствии с ТЗ;
4. Выгрузка отчетов по успеваемости, посещаемости и т.д.;
5. Функции по просмотру статистики, описанной в ТЗ;
6. Функции по выгрузке отчетов из разделов, описанных в ТЗ.

Ученикам доступны следующие функции:

1. Добавление данных по разделам, описанным в ТЗ;
2. Удаление данных по разделам, описанным в ТЗ;
3. Просмотр статистики по персональной успеваемости, посещаемости и т.д.;
4. Выгрузка отчетов по персональной успеваемости.

## Цель работы и задачи

**Цель работы:** спроектировать диаграмму классов, помогающей реализовать модуль корпоративно-информационной системы онлайн-школы *«Exam Masters».*

**Задачи:**

* + - 1. Выбрать средство реализации диаграммы классов;
      2. Согласно ТЗ определить необходимые в системе классы;
      3. Определить связи между классами и компонентами;
      4. Разработать диаграмму классов;
      5. Составить описание диаграммы классов.

# Диаграмма классов

## Средство разработки

Для разработки диаграммы классов было принято использовать бесплатный онлайн-сервис *«Draw.io»*. Данный онлайн-сервис предназначен для создания диаграмм, блок-схем, схем бизнес-процессов, *UML*, схем *ER*-моделей и т.д.

Достоинство сервиса заключается в его обширном функционале. Иными словами, пользователь может работать с большим набором форм, стрелок и других элементов, необходимым для создания диаграмм. Помимо того, в сервисе также предусмотрены функции экспорта и импорта изображений в различных форматах: *PDF*, *XML*, *JPG*, *PNG* и т.д.

Общий алгоритм работы с сервисом представлен ниже:

Создание новой диаграммы. На данном этапе можно выбрать как пустую диаграмму, так и определенный тип диаграммы, заложенный в систему.

Добавление элементов. После создания диаграммы можно добавлять элементы на полотно, перетаскивая их из панели инструментов.

Сохранение и экспорт. После завершения проектирования диаграммы можно сохранить схему либо в облачном хранилище, либо на компьютере в любом удобном формате.

Совместная работа. Сервис позволяет совместно проектировать схемы и диаграммы. Пользователь может пригласить других пользователей к проекту для просмотра и редактирования.

## 

## Описание диаграммы классов

На Рис. 1 ниже показана диаграмма классов, спроектированная в сервисе *«Draw.io»*:

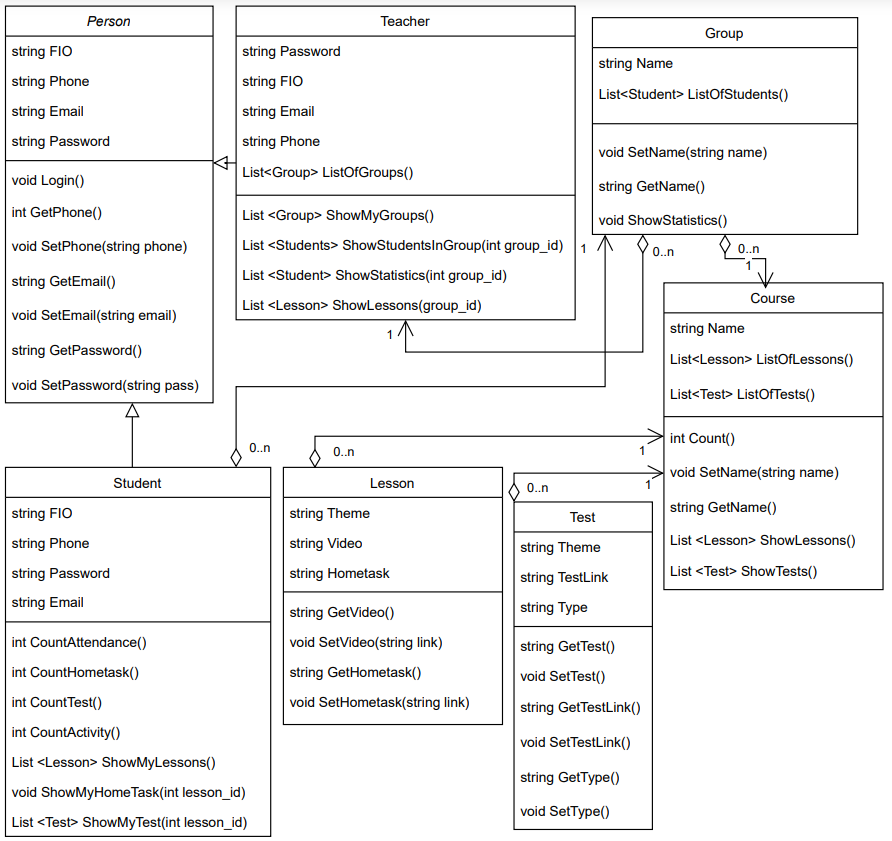


Рис. . Диаграмма классов для реализации КИС «Exam Masters»

Таким образом, в системе будут реализованы 7 классов. Далее следует описать классы и их компоненты, которые будут предусмотрены в системе при дальнейшей разработке.

**Класс «*Person».*** Класс используется в самом начале в окне авторизации. Ниже в Табл. 1 представлены поля класса «Person»:

Табл. . Поля класса «Person»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| FIO | String | ФИО пользователя |
| Phone | String | Номер телефона пользователя |
| Email | String | Электронная почта пользователя (в системе также подразумевается, как логин при входе в систему) |
| Password | String | Пароль от учетной записи пользователя |

Далее были прописаны методы класса «Person». Более подробное описание представлено ниже на Табл. 2:

Табл. . Методы класса «Person»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| Login | Void | Метод предназначен для авторизации пользователя в системе. |
| GetPhone | String | Метод возвращает исходное значение номера телефона пользователя. |
| SetPhone | String | Метод **на вход** принимает строку.  Метод присваивает новое значение полю Phone класса Person. |
| GetEmail | String | Метод возвращает исходное значение электронной почты пользователя. |
| SetEmail | String | Метод **на вход** принимает строку.  Метод присваивает новое значение полю Email класса Person. |
| GetPassword | String | Метод возвращает исходное значение пароля пользователя для входа в систему. |
| SetPassword | String | Метод **на вход** принимает строку.  Метод присваивает новое значение полю Password класса Person. |

**Класс «*Teacher».*** Класс наследует поля и методы класса «Person». Ниже в Табл. 3 представлены поля класса «Teacher»:

Табл. . Поля класса «Teacher»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| FIO | String | ФИО преподавателя |
| Phone | String | Номер телефона преподавателя |
| Email | String | Электронная почта преподавателя |
| Password | String | Пароль от учетной записи преподавателя |
| ListOfGroups | List <Group> | Список групп, которые ведет преподаватель.  *В данном случае, используется тип данных «Group», который является отдельным классом в системе. Он будет описан ниже.* |

Так как класс «Teacher» является дочерним от класса «Person», он наследует и методы материнского класса. Помимо этого, дочерний класс выполняет специфические методы, которые описаны ниже на Табл. 4:

Табл. . Методы класса «Teacher»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| ShowMyGroups | List <Group> | Метод возвращает список групп, в которых преподаватель ведет занятия. |
| ShowStudentsInGroup | List <Student> | **На** **вход** метод получает идентификатор группы.  Возвращает список студентов, которые принадлежат группе, в которой преподаватель ведет занятия. |
| ShowStatistics | List <Group> | **На** **вход** метод получает идентификатор группы.  Метод выводит статистику по ученикам группы, в которой преподаватель ведет занятия. |
| ShowLessons | List <Lesson> | **На** **вход** метод получает идентификатор группы.  Метод выводит занятия группы, в которой преподаватель ведет занятия. |

**Класс «*Student».*** Класс наследует поля и методы класса «Person». Ниже в Табл. 5 представлены поля класса «Student»:

Табл. . Поля класса «Student»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| FIO | String | ФИО ученика |
| Phone | String | Номер телефона ученика |
| Email | String | Электронная почта ученика |
| Password | String | Пароль от учетной записи ученика |

Так как класс «Student» является дочерним от класса «Person», он наследует и методы материнского класса. Помимо этого, дочерний класс выполняет специфические методы, которые описаны ниже на Табл. 6:

Табл. . Методы класса «Student»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| CountAttendance | Int | Метод возвращает процент посещаемости занятий по предмету. |
| CountHometask | Int | Метод возвращает процент домашнего задания по предмету. |
| CountTest | Int | Метод возвращает среднюю оценку за контрольные точки по предмету. |
| CountActivity | Int | Метод возвращает среднюю оценку за активность на занятиях по предмету. |
| ShowMyLessons | List <Lesson> | Метод возвращает список уроков на курсе. |
| ShowMyHomeTask | Void | Метод возвращает список домашнего занятия на курсе. |
| ShowMyTest | List <Test> | Метод возвращает список тестов на курсе. |

**Класс «*Group».*** Класс является объектом-агрегатом класса «Teacher». Таким образом, была реализована связь «один-ко-многим». В свою очередь, между классом «Group» и «Student» также реализована эта связь, так как в группе может быть несколько учеников. Ниже в Табл. 7 представлены поля класса «Group»:

Табл. . Поля класса «Group»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| Name | String | Наименование группы |
| ListOfStudents | List <Student> | Список учеников, которые обучаются в одной группе |

Далее были прописаны методы, которые присущи только классу «Group». Эти методы описаны ниже на Табл. 8:

Табл. . Методы класса «Group»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| SetName | Void | Метод **на вход** принимает строку и присваивает новое значение полю «Name». |
| GetName | String | Метод возвращает значение поля «Name». |
| ShowStatistics | String | Метод возвращает статистику по группе. |

**Класс «*Course».*** Между этим классом и классом «Croup» также реализована связь «один-ко-многим», так как несколько разных групп могут заниматься по программе одного курса. Ниже в Табл. 9 представлены поля класса «Course»:

Табл. . Поля класса «Course»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| Name | String | Наименование курса |
| ListOfLessons | List <Lesson> | Список уроков на курсе |
| ListOfTests | List <Test> | Список тестов на курсе |

Далее были прописаны методы, которые присущи только классу «Course». Эти методы описаны ниже на Табл. 10:

Табл. . Методы класса «Course»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| Count | Int | Метод подсчитывает количество уроков на курсе. |
| SetName | Void | Метод присваивает новое значение наименования курса. |
| GetName | String | Метод возвращает наименование курса. |
| ShowLessons | List <Lesson> | Метод возвращает список уроков курса. |
| ShowTests | List <Test> | Метод возвращает список тестов курса. |

**Класс «*Lesson».*** Класс является объектом-агрегатом класса «Group». Таким образом, была реализована связь «один-ко-многим», так как в одном курсе может быть много занятий. Ниже в Табл. 11 представлены поля класса «Lesson»:

Табл. . Поля класса «Lesson»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| Theme | String | Тема занятия |
| Video | String | Ссылка для подключения занятию |
| Hometask | String | Ссылка домашнего задания к занятию |

Далее были прописаны методы, принадлежащие классу «Lesson». Эти методы описаны ниже на Табл. 12:

Табл. . Методы класса «Lesson»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| SetVideo | Void | Метод принимает **на вход** ссылку и присваивает новое значение ссылки для подключения к занятию. |
| GetVideo | String | Метод возвращает ссылку для подключения к занятию. |
| SetHometask | Void | Метод принимает **на вход** ссылку и присваивает новое значение ссылки для выполнения домашнего задания к занятию. |
| GetHometask | String | Метод возвращает ссылку на домашнее задание. |

**Класс «*Test».*** Класс является объектом-агрегатом класса «Group». Таким образом, была реализована связь «один-ко-многим», так как в одном курсе может быть несколько тестов. Ниже в Табл. 13 представлены поля класса «Test»:

Табл. . Поля класса «Test»

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Описание** |
| Theme | String | Тема контрольной точки |
| TestLink | String | Ссылка для прохождения контрольной точки |
| Type | String | Тип контрольной точки |

Далее были прописаны методы, принадлежащие классу «Test». Эти методы описаны ниже на Табл. 14:

Табл. . Методы класса «Test»

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип данных** | **Описание** |
| GetTheme | string | Метод возвращает название контрольной точки. |
| SetTheme | Void | Метод принимает **на вход** строку и присваивает новое значение темы теста. |
| GetTestLink | string | Метод возвращает ссылку для прохождения контрольной точки. |
| SetTestLink | Void | Метод принимает **на вход** строку и присваивает новое значение ссылки для прохождения контрольной точки. |
| GetType | string | Метод возвращает тип контрольной точки. |
| SetType | Void | Метод принимает **на вход** строку и присваивает новое значение типа контрольной точки. |

# Выводы

В ходе настоящей лабораторной работы была спроектирована диаграмма классов для корпоративной информационной системы онлайн-школы *«Exam Masters».* Были реализованы необходимы в системе классы, а также продуманы из поля и методы. Помимо этого, были реализованы наиболее рациональные связи между классами согласно ТЗ и хранилищу данных.

В результате выполнения лабораторной работы была спроектирована диаграмма классов, которая полностью покрывает все требования ТЗ и обеспечивает наиболее удобную дельнейшую реализацию системы. В структуре классов были учтены основные составляющие, задействованные также в хранилище данных. Помимо того, методы классов были подобраны таким образом, чтобы можно было реализовать все функции, указанные в ТЗ.

Согласно ТЗ следующим этапом разработки информационной системы является реализация программного кода.