# Отчёт по разработке прикладных приложений:

Работу выполнил ученик группы 19ПМИ-1: Кабанов Денис

Ссылка на git-hub репозиторий с проектом, описанием требований и тестированием: <https://github.com/denis-kabanov/Applicationd_development>

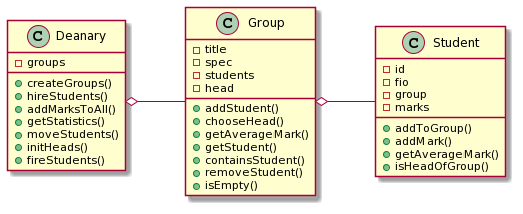
Главной целью проекта являлось написание простой модели деканата, со следующими требованиями:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **Источник** | **Группа** | **Требование** | **Детальное требование** |
| 1 | Заказчик | Функциональные | Учет студентов | Наличие полей id (идентификатор), FIO (ФИО), group (ссылка на группу), marks (оценки) у объекта-студента |
| 2 | Заказчик | Функциональные | Учет студентов | Создание студента с указанием ИД и ФИО |
| 3 | Заказчик | Функциональные | Учёт студентов | Зачисление в группу |
| 4 | Заказчик | Функциональные | Учёт студентов | Добавление оценки |
| 5 | Заказчик | Функциональные | Учёт групп студентов | Наличие полей title (название группы), spec (специльность), students(вектор студентов) head (староста) |
| 6 | Заказчик | Функциональные | Учёт групп студентов | Создание группы с указанием названия и специальности |
| 7 | Заказчик | Функциональные | Учёт групп студентов | Возможность добавления студента в группу |
| 8 | Заказчик | Функциональные | Учёт групп студентов | Избрание старосты |
| 9 | Заказчик | Функциональные | Учёт групп студентов | Поиск студентов по ФИО или id |
| 10 | Заказчик | Функциональные | Учёт групп студентов | Вычисление среднего балла в группе |
| 11 | Заказчик | Функциональные | Учёт групп студентов | Исключение студента из группы |
| 12 | Заказчик | Функциональные | Объект-деканат | Содержит массив групп |
| 13 | Заказчик | Функциональные | Объект-деканат | Создание групп и заполнение их студентами на основе файла |
| 14 | Заказчик | Функциональные | Объект-деканат | Добавление случайных оценок студентам |
| 15 | Заказчик | Функциональные | Объект-деканат | Получение статистики по успеваемости студентов и групп |
| 16 | Заказчик | Функциональные | Объект-деканат | Отчисление студентов за неуспеваемость |
| 17 | Заказчик | Функциональные | Объект-деканат | Инициация выборов старост в группах |
| 18 | Пользователи | Нефункциональные | Ограничение доступа | Доступ разрешён всем пользователям git-hub (opensource) |
| 19 | Пользователи | Нефункциональные | Ограничение доступа | Возможность для скачивания в любой момент |

## **Структура проекта**

* **include/Student.h** - заголовочный файл для класса **Student**.
* **include/Group.h** - заголовочный файл для класса **Group**.
* **include/Deanery.h** - заголовочный файл для класса **Deanery**.
* **src/Student.cpp** - исходный файл для класса **Student**.
* **src/Group.cpp** - исходный файл для класса **Group**.
* **src/Deanery.cpp** - исходный файл для класса **Deanery**.
* **src/main.cpp** - демонстрационная программа.

UML диаграмма классов:



Вывод: в этом отчёте были представлены основные требования к проекту, как функциональные, так и нефункциональные, все они были выполнены. Основные функциональные требований покрыты тестами и/или их работа предоставлена в файле main.cpp.