МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Проектирование информационных систем**

**Тема:** «Объектно-ориентированное моделирование. СТРУКТУРНЫЕ ДИАГРАММЫ UML»

**Цель:** Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования структуры информационной системы с применением UML.

Выполнил:

Ильин Н. С.,

3 курс 7 группа

Проверил:

Якунович А. В.

Минск 2023

**Описание программных средств**

Draw.io — инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

Инструмент работает с Google Диск, Google Workspace и Dropbox, глубоко интегрирован и удобен для работы с продуктами Confluence и Jira от Atlassian. Пользователи также могут работать с диаграммами в автономном режиме и сохранять их локально, используя настольное приложение для персональных компьютеров.

Инструмент позволяет создавать: графики, диаграммы, таблицы, презентации, блок-схемы, планы помещений, воронки продаж, ментальные карты, карты сайтов.

**Постановка задачи**

Задачей данной работы выступает описание структуры взаимодействия классов в приложении, описанной в лабораторной работе #1.

Главными сущностями приложения выступают:

* AppContext – класс с помощью которого осуществляется доступ к базе данных данного приложения.
* AuthContoller – осуществляет обработку запросов, предназначенных пользователю. А именно: регистрация, авторизация, выход из аккаунта.
* SubjController – осуществляет обработку запросов, предназначенных предметам. А именно: добавление и удаление предмета.
* LessonController – осуществляет обработку запросов, предназначенных длязанятий. А именно: добавление, изменение, удаление занятий.
* HWController – осуществляет обработку запросов, предназначенных для домашнего задания. А именно: добавление и удаление ДЗ.
* SchController – осуществляет обработку запросов, предназначенных для расписания. А именно: добавление, удалениезанятия в расписании и изменение расписания.
* SearchFilterController – осуществляет обработку запросов, предназначенных поиску и фильтрации. А именно: установке фильтров иих применение**.**

**Описание практического задания**

Разработанная диаграмма классов представлена на рисунке 1.1.

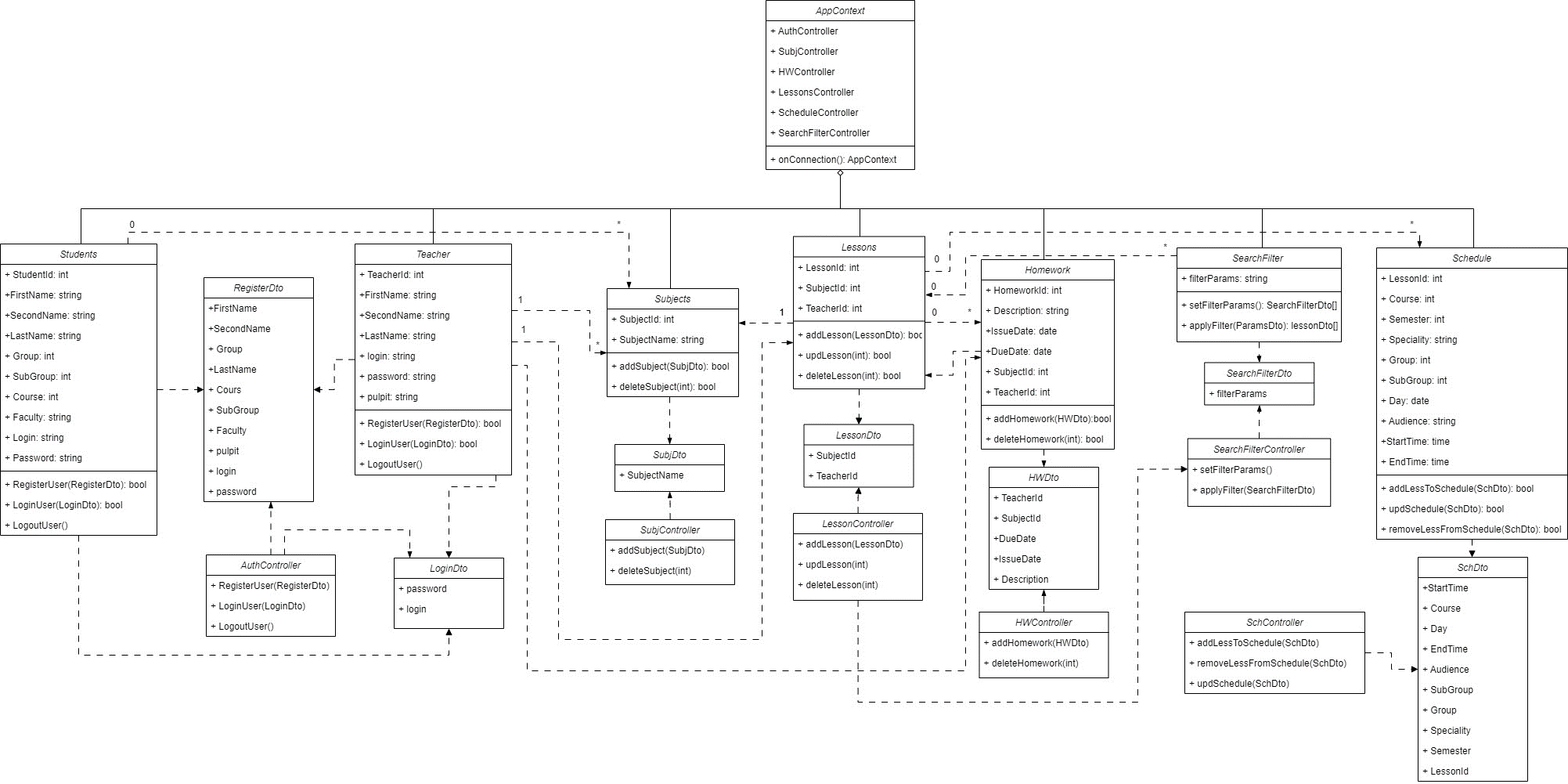


Рисунок 1.1 – Диаграмма классов приложения расписания занятий

1. Между классами «Преподаватель» (Teacher) и «Предмет» (Subject):
   * Кратность: 1..\* (Один преподаватель может преподавать один или несколько предметов, но каждый предмет должен иметь хотя бы одного преподавателя.)
2. Между классами «Студент» (Student) и «Предмет» (Subject):
   * Кратность: 0..\* (Студент может выбирать ноль или более предметов для изучения, а каждый предмет может быть выбран несколькими студентами.)
3. Между классами «Занятие» (Lesson) и «Преподаватель» (Teacher):
   * Кратность: 1 (Каждое занятие должно иметь одного и только одного преподавателя, а преподаватель может вести несколько занятий.)
4. Между классами «Занятие» (Lesson) и «Предмет» (Subject):
   * Кратность: 1 (Каждое занятие должно быть связано с одним и только одним предметом, а предмет может иметь несколько занятий.)
5. Между классами «Домашнее задание» (Homework) и «Занятие» (Lesson):
   * Кратность: 0..\* (Занятие может иметь ноль или более домашних заданий, а каждое домашнее задание должно быть связано с одним занятием.)
6. Между классами «Расписание» (Schedule) и «Занятие» (Lesson):
   * Кратность: 0..\* (Расписание может содержать ноль или более занятий, а каждое занятие должно быть связано с одним расписанием.)
7. Между классами «Поиск и фильтрация» (Search and Filter) и «Занятие» (Lesson):
   * Кратность: 0..\* (Поиск и фильтрация может быть связан с нулем или более занятиями, а каждое занятие может быть связано с поиском и фильтрацией.)

AppContext – класс с помощью которого осуществляется доступ к базе данных данного приложения.

Этот класс имеет следующие методы:

* + OnConnect – защищенный метод, который выполняется при первом получении доступа к БД.

Далее более подробно будут описаны каждый из модулей диаграммы.

1. ***Модуль «User»:***

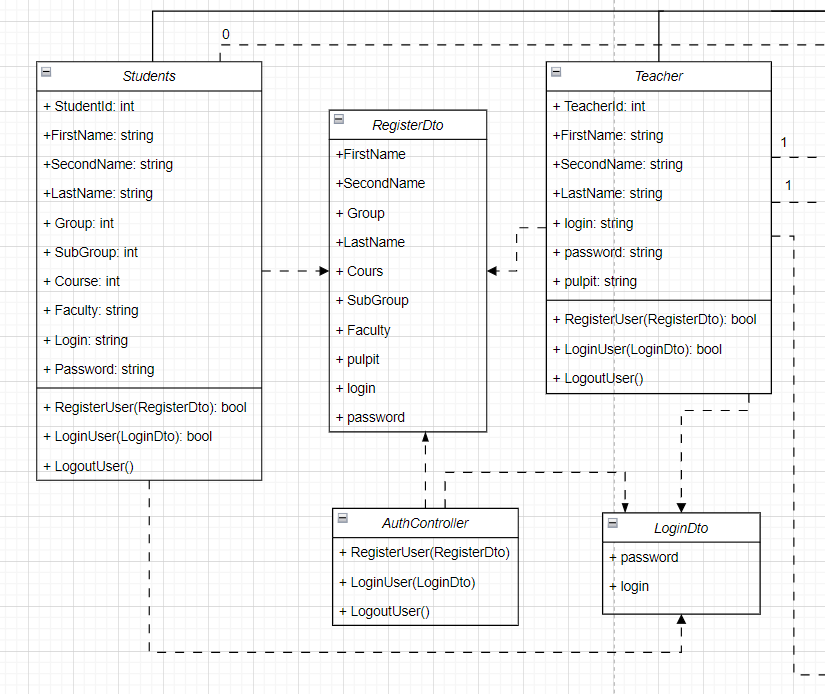


Рисунок 1.2 – Модуль «Users»

AuthController – осуществляет обработку запросов, предназначенных пользователю. А именно: регистрация, авторизация и выход из аккаунта. Он содержит только публичные методы:

* + LoginUser – обработка запроса на авторизацию;
  + RegisterUser – обработка запроса на регистрацию;
  + LogoutUser – обработка запроса на выход из аккаунта;

Teacher – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера AuthController. Далее перечислены все публичные свойства:

* + TeacherId – id преподавателя;
  + FirstName– имя преподавателя;
  + SecondName – фамилия преподавателя;
  + LastName – отчество преподавателя;
  + login – логин преподавателя;
  + password – пароль преподавателя;
  + pulpit – кафедра.

Students – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера AuthController. Далее перечислены все публичные свойства:

* + StudentId – id студента;
  + FirstName– имя студента;
  + SecondName – фамилия студента;
  + LastName – отчество студента;
  + Сourse – курс;
  + Faculty – факультет;
  + login – логин студента;
  + password – пароль студента;
  + SubGroup – подгруппа;
  + Group – группа.

Далее указаны публичные методы класса Teacher и класса Students:

* RegisterUser – выполняет добавление данных пользователя в бд;
* LoginUser – выполняет проверку, зарегистрирован ли данный пользователь;
* Logout – выполняет выход из аккаунта;

LoginDto - класс, который используется для получение данных, необходимых при авторизации. Содержит только публичные свойства:

* login – ник пользователя;
* Password – пароль пользователя.

RegisterDto – класс, который используется для получение данных, необходимых при регистрации. Содержит только публичные свойства:

* login – ник пользователя;
* Password – пароль пользователя.

1. ***Модуль «Subjects»:***

SubjController – осуществляет обработку запросов, предназначенных предметам. А именно: добавление и удаление предмета. Он содержит только публичные методы:

* + addSubject – обработка запроса на добавление предмета;
  + deleteSubject – обработка запроса на удаление предмета.

Subjects – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера SubjController.

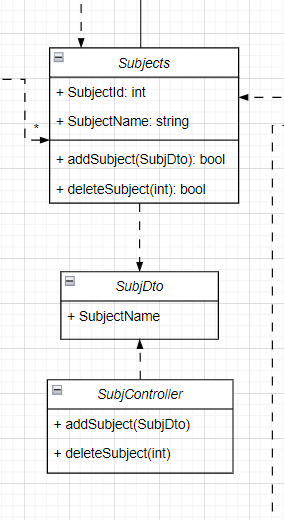


Рисунок 1.3 – Модуль «Subjects»

Далее перечислены все публичные свойства класса Subject:

* + SubjectId – id предмета;
  + SubjectName – название предмета.

Далее указаны публичные методы класса Posts:

* + addSubject – выполняет добавление предмета;
  + deleteSubject – выполняет удаление предмета.

SubjDto – класс, который используется для получение данных, необходимых для действий с предметами. Содержит только публичные свойства:

* + SubjectName – название предмета.

1. ***Модуль «Lessons»:***

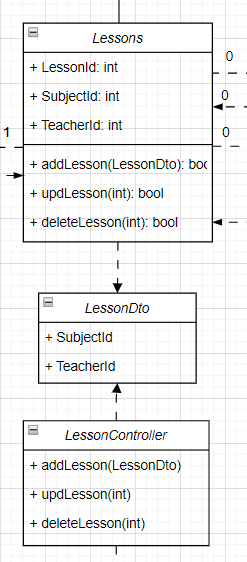


Рисунок 1.4 – Модуль «Lessons»

LessonController – осуществляет обработку запросов, предназначенных для занятий. А именно: добавление, изменение, удаление занятий. Он содержит только публичные методы:

* + addLesson – обработка запроса на добавление занятий;
  + updLesson – обработка запроса на изменение занятий;
  + deleteLesson – обработка запроса на удаление занятий.

Lessons – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера LessonController.

Далее перечислены все публичные свойства:

* + LessonId – id занятия;
  + SubjectId – id предмета;
  + TeacherId – id преподавателя.

Далее указаны публичные методы:

* + addLesson – выполняет добавление занятий;
  + updLesson – выполняет изменение занятий;
  + deleteLesson – выполняет удаление занятий.

LessonDto – класс, который используется для получение данных, необходимых для действий с занятиями. Содержит только публичные свойства:

* + SubjectId – id предмета;
  + TeacherId – id преподавателя.

1. ***Модуль «Homework»:***

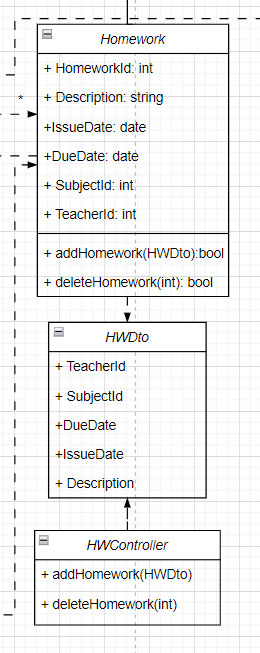


Рисунок 1.5 – Модуль «Homework»

HWController – осуществляет обработку запросов, предназначенных для домашнего задания. А именно: добавление и удаление ДЗ. Он содержит только публичные методы:

* + addHomework – обработка запроса на добавление ДЗ;
  + deleteHomework – обработка запроса на удаление ДЗ.

Homework – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера HWController.

Далее перечислены все публичные свойства:

* + HomeworkId – id ДЗ;
  + Description – описание ДЗ;
  + IssueDate – дата выдачи;
  + DueDate – крайник срок выполнения;
  + SubjectId – id предмета;
  + TeacherId – id преподавателя.

Далее указаны публичные методы:

* + addHomework – выполняет добавление ДЗ;
  + deleteHomework – выполняет удаление ДЗ.

HWDto – класс, который используется для получение данных, необходимых для ДЗ. Содержит только публичные свойства:

* + Description – описание ДЗ;
  + IssueDate – дата выдачи;
  + DueDate – крайник срок выполнения;
  + SubjectId – id прелмета;
  + TeacherId – id преподавателя.

1. ***Модуль «Schedule»:***

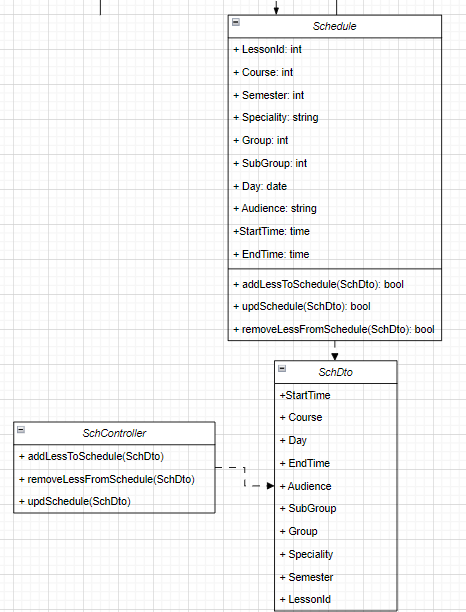


Рисунок 1.6 – Модуль «Schedule»

SchController – осуществляет обработку запросов, предназначенных для расписания. А именно: добавление, удалениезанятия в расписании и изменение расписания. Он содержит только публичные методы:

* + addLessToSchedule – обработка запроса на добавление занятия в расписание;
  + updSchedule – обработка запроса на изменение расписания;
  + removeLessFromSchedule – обработка запроса на удаление занятия из расписания.

Schedule – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера SchController.

Далее перечислены все публичные свойства:

* + LessonId – id занятия;
  + Semester – семестр;
  + Speciality – специальность;
  + Course – курс;
  + Group – группа;
  + SubGroup – подгруппа;
  + Day – день;
  + StartTime – время начала пары;
  + EndTime – время конца пары;
  + Audienсe – аудитория.

Далее указаны публичные методы:

* + addLessToSchedule – выполняет добавление занятия в расписание;
  + updSchedule – выполняет изменение расписания;
  + removeLessFromSchedule – выполняет удаление занятия из расписания.

SchDto – класс, который используется для получение данных, необходимых для расписания. Содержит только публичные свойства:

* + LessonId – id занятия;
  + Semester – семестр;
  + Speciality – специальность;
  + Course – курс;
  + Group – группа;
  + SubGroup – подгруппа;
  + Day – день;
  + StartTime – время начала пары;
  + EndTime – время конца пары;
  + Audienсe – аудитория.

1. ***Модуль «SearchFilter»:***

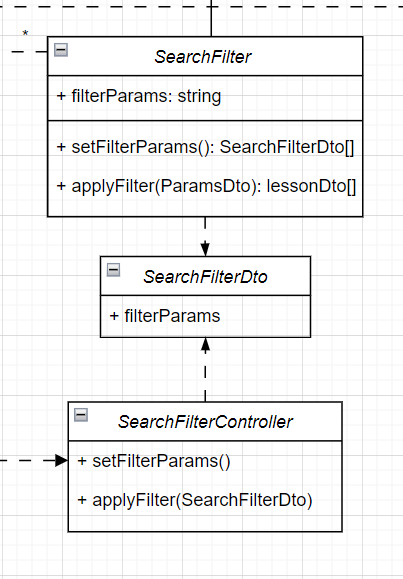


Рисунок 1.7 – Модуль «SearchFilter»

SearchFilterController – осуществляет обработку запросов, предназначенных для поиска и фильтрации. А именно: установка фильтров и их применение. Он содержит только публичные методы:

* + setFilterParams – обработка запроса на выбор параметров;
  + applyFilter – обработка запроса на применение фильтрации.

SearchFilter – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера SearchFilterController и PostController.

Далее перечислены все публичные свойства:

* + filterParams – параметры фильтра.

Далее указаны публичные методы:

* + setFilterParams –выполняет выбор параметров;
  + applyFilter – выполняет применение фильтрации.

SearchFilterDto – класс, который используется для получение данных, необходимых для параметров фильтрации. Содержит только публичные свойства:

* + filterParams – параметры фильтра.

Далее представлены пакеты, в которые были сгруппированы классы.

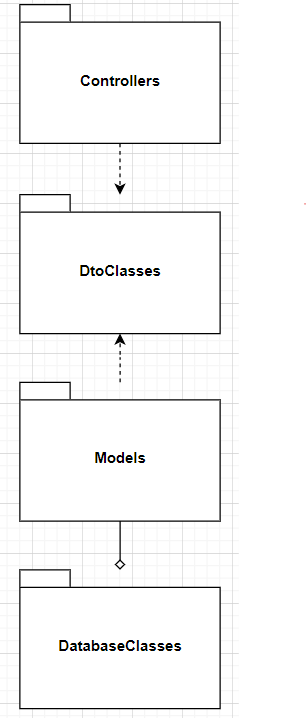


Рисунок 1.8 – Пакеты классов

Controllers содержит классы, отвечающие за обработку запросов на сервер. К ним относятся: AuthController, SubjController, LessonController, HWController, SchController, SearchFilterController.

DtoClasses содержит классы, отвечающие за передаваемые в контроллерах данные. К этим классам относятся: LoginDto, RegisterDto, SubjDto, LessonDto, HWDto, SchDto, SearchFilterDto.

Models содержит классы, отвечающие за взаимодействие с базой данных. К этим классам относятся: Students, Teacher, Subjects, Lessons, Homework Schedule, SearchFilter .

DatabaseClasses содержит классы, которые позволяют организовать соединение с бд. К данным классам относится AppContext.

**Вывод:** Я изучил методологию объектно-ориентированного моделирования средствами UML. Ознакомился с основными принципами объектно-ориентированного проектирования программного обеспечения, получил навыки проектирования функциональности информационной системы с применением UML.