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: **«Разработка структур классов»**

Выполнили:

студенты гр. ИСТ-19-1Б

Ким Елизавета Алексеевна

Решетникова Елизавета Андреевна

Лопина Мария Сергеевна

(Ф.И.О.)

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

*(подпись)*

Принял:

ст. преподаватель, Банников Р.Ю.

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

*(должность, ФИО руководителя)*

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

*(оценка) (подпись)*

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

*(дата)*

**Пермь 2023**

Оглавление

[**1. Постановка задачи 3**](#_heading=h.suk4hvi05v6g)

[Описание назначения КИС и основного функционала согласно ТЗ. 3](#_heading=h.2np5qgqz6afq)

[Возможности системы 3](#_heading=h.tirl2i5o87xu)

[Цели создания системы 3](#_heading=h.bft03p97uy5k)

[Текущее состояние объекта автоматизации 3](#_heading=h.lourjbq4p286)

[Функционал ролей пользователей 3](#_heading=h.qi0xnvjkcw3d)

[Цель работы и задачи, которые нужно выполнить для достижения цели: 4](#_heading=h.j5tid6le9gk)

[**2. Разработка диаграммы классов 4**](#_heading=h.ia3hy9wqroz5)

[A. Выбор средства и среду для разработки диаграммы классов. 4](#_heading=h.cduaubr5hkb8)

[B. Проектирование диаграммы классов. 4](#_heading=h.omzx43su5e0c)

[C. Назначdvrstние классов и их методов. 7](#_heading=h.i3jelwc62aob)

[**3. Вывод 18**](#_heading=h.2bsmv0ahhako)

[**4. Приложение 19**](#_heading=h.o0ys6zwsfx7u)

[Приложение 1. Диаграмма класса АИС “UpsilonConnect” 19](#_heading=h.v2ek8a2nkarv)

# 1. Постановка задачи

## Описание назначения КИС и основного функционала согласно ТЗ.

Автоматизированная информационная система предназначена для автоматизации поиска тем для научных работ и Соискателей для выполнения задач, поставленных Работодателями.

### Возможности системы

* Сбор в одной информационной системе задач, представленных организациями для взаимодействия с исполнителями.
* Создание ведение карточек-задач со стороны Работодателей с подробной информацией о поставленной задаче.
* Создание и ведение карточек-анкет со стороны соискателей о имеющихся навыках и опыте.

### Цели создания системы

* Ведение базы данных с доступными задачами для их выполнения.
* Ведение базы данных с резюме Соискателей.
* Управление процессами взаимодействия между Соискателем и Работодателем.

### Текущее состояние объекта автоматизации

На текущий момент взаимодействие работодателя и исполнителя по поиску и выбору задачи осуществляется вручную без фиксации контроля со стороны работодателя действий исполнителя. В связи с чем данные получаются некорректными.

### Функционал ролей пользователей

Роли:

А) Администратор

Б) Соискатель

В) Работодатель

Функционал ролей:

Администратор:

1. редактирование пользователей;
2. просмотр списка пользователей;
3. удаление пользователей;
4. редактирование прав доступа для пользователя;
5. блокировка пользователей.

Соискатель:

1. редактирование личной информации профиля;
2. размещение резюме;
3. просмотр списка опубликованных задач;
4. отклик на задачу;

Работодатель:

1. редактирование публичной информации профиля;
2. размещение задачи;
3. просмотр резюме Соискателей;
4. подтверждение отклика от Соискателя;

## Цель работы и задачи, которые нужно выполнить для достижения цели:

Цель: диаграмма классов для интерфейсной и функциональной части АИС, удовлетворяющий функционалу, описанному в ТЗ.

Задачи:

1. Выбрать средства и среду для разработки диаграммы классов;
2. Спроектировать диаграмму классов (форму авторизации, вид приложения под разными пользователями, редактирование информации в БД, отчеты и реакция на ошибочный ввод);
3. Описать назначение классов и их методов.

# 2. Разработка диаграммы классов

## Выбор средства и среду для разработки диаграммы классов.

Диаграмма классов - это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними. Она широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Архитектура была спроетирована в Diagrams.net (ранее draw.io) - это бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс используется для создания таких диаграмм, как блок-схемы, каркасы, UML-диаграммы, организационные диаграммы и сетевые диаграммы.

## Проектирование диаграммы классов.

Для разработки архитектуры использовался паттерн MVC (рис.1).

**Model-View-Controller** (**MVC**, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — это архитектурный паттерн, который разделяет данные приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Паттерн помогает грамотно организовать собственный код — по функциям каждого конкретного блока. Повышает читаемость, модернизацию и поддержку кода.

* ***Представление*** (*View*) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
* ***Модель*** (*Model*) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние.
* ***Контроллер*** (*Controller*) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

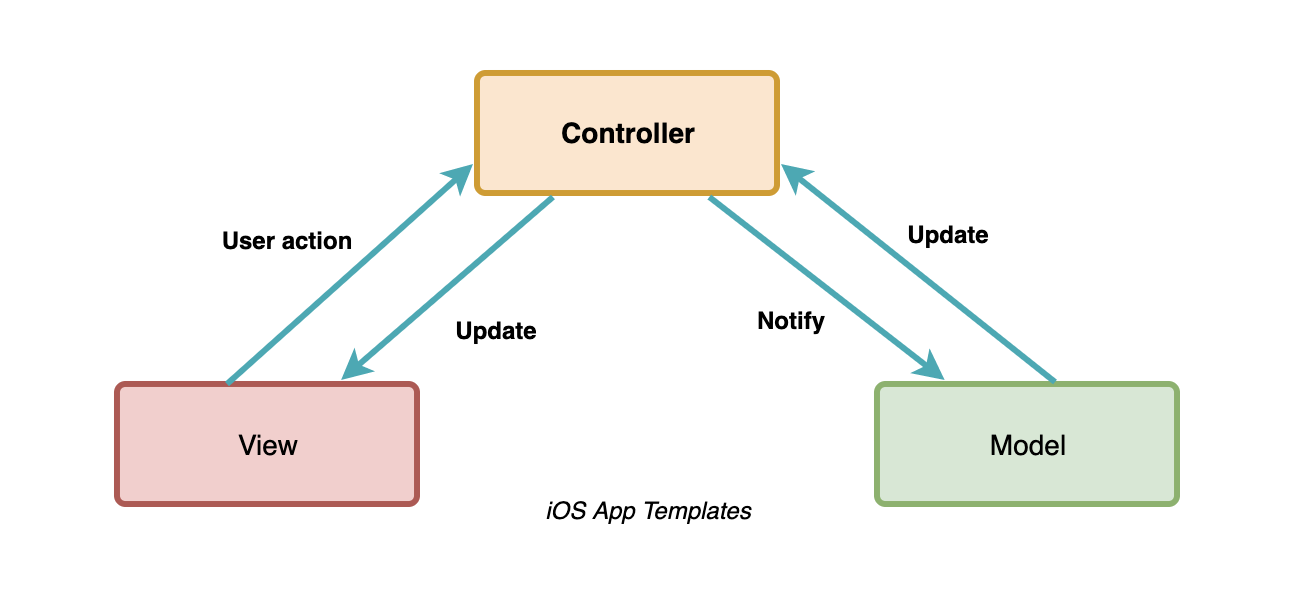


Рис.1-Модель MVC

Была разработана диаграмма классов для АИС “UpsilonConnect” (приложение 1). На диаграмме отображен функционал системы, а именно авторизация, регистрация и выход пользователей, взаимодействие с страницами Задачи, Резюме и личный профиль пользователя, формирование отчета. Далее компоненты паттерна рассмотрены по отдельности. Представление - рисунок 2, контроллер - рисунок 3, модель - рисунок 4.

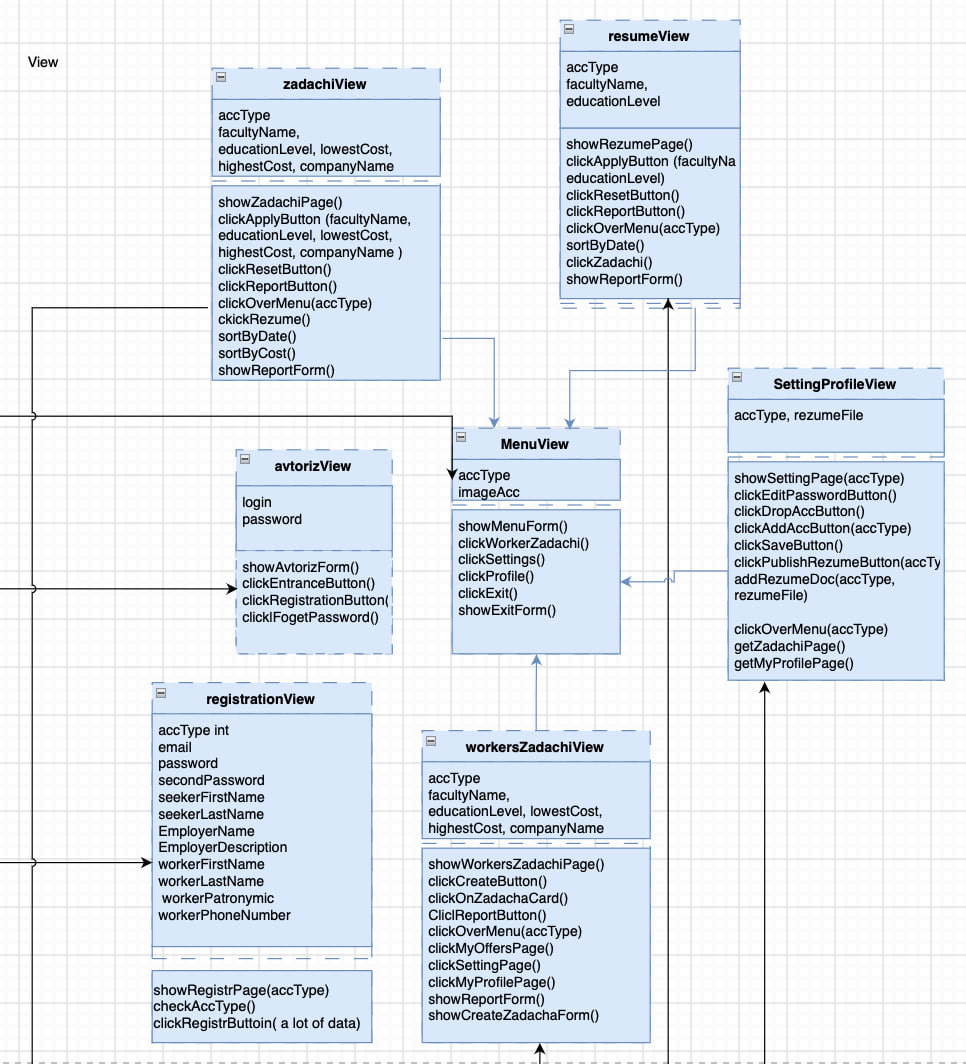


Рис.2 - View

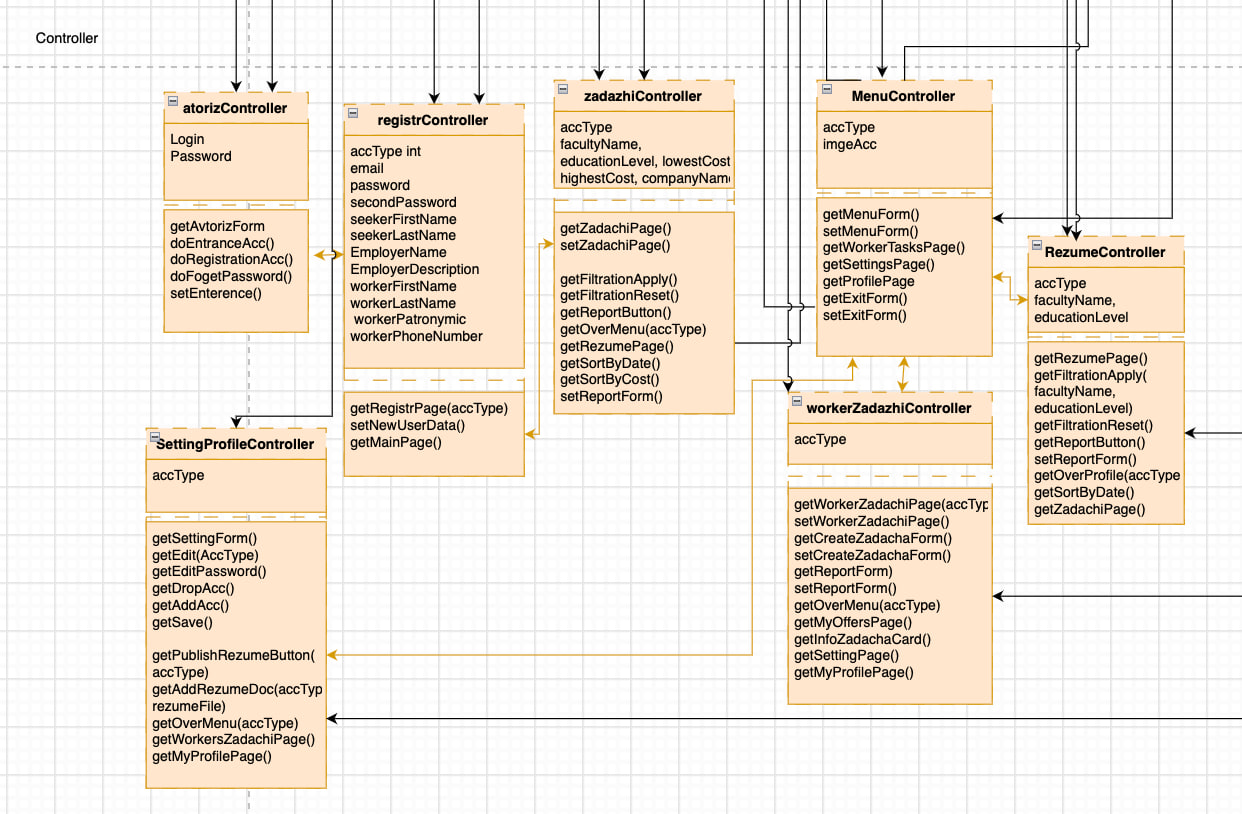


Рис.3 - Controller

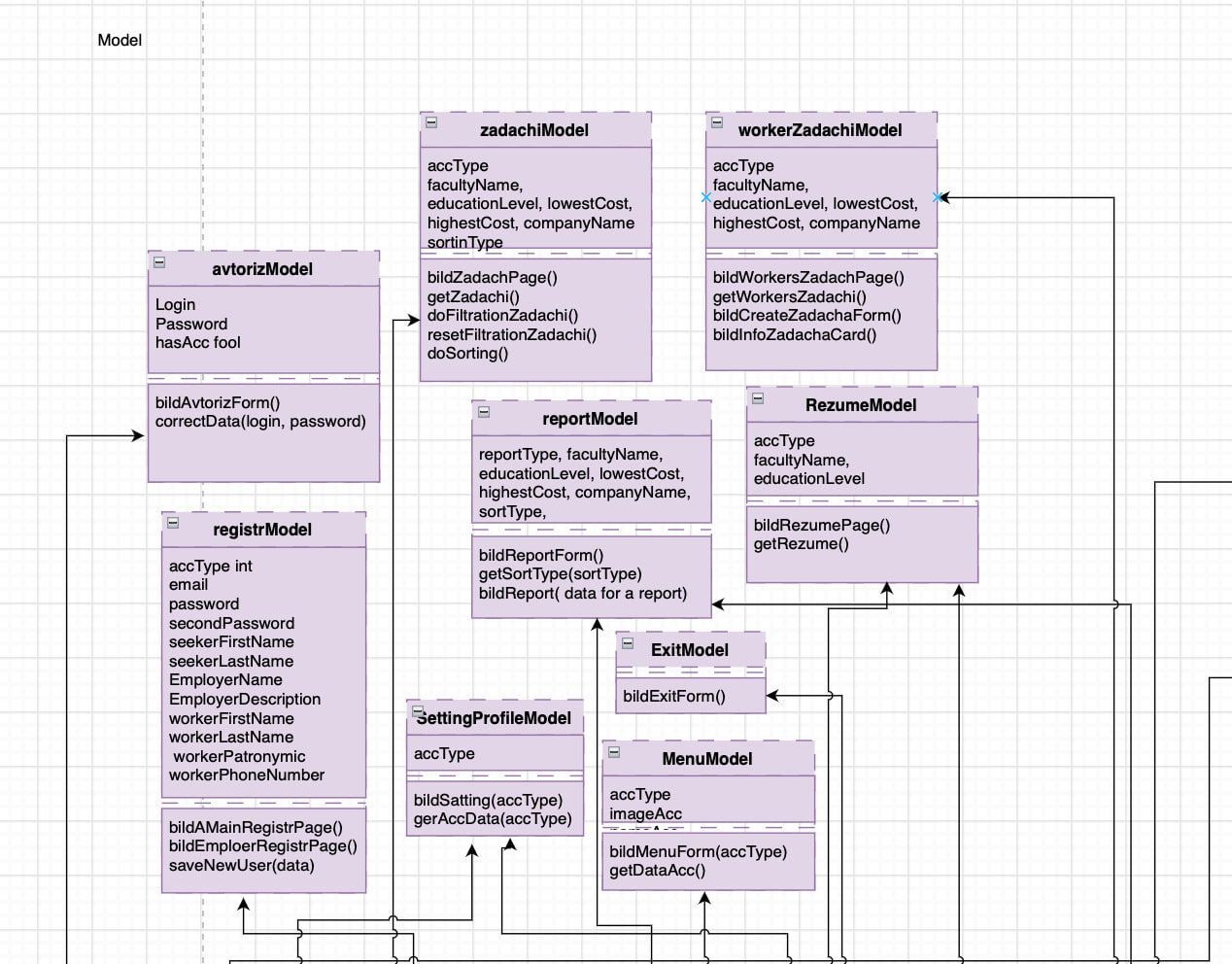


Рис.4 - Model

## Назначние классов и их методов.

В таблице 1 описаны назначения классов и их методов разрабатываемой системы.

Таблица 1 Классы и методы системы UpsilonConnect

| **Имя класса** | **Поле класса** | **Назначение класса** | **Название метода** | **Назначение метода** |
| --- | --- | --- | --- | --- |
| registrationView | accType int  email  password  secondPassword  seekerFirstName  seekerLastName  EmployerName  EmployerDescription  workerFirstName  workerLastName  workerPatronymic  workerPhoneNumber | класс описывает взаимодействие представление регистрации | showRegistrPage(accType) | запрашивает у контроллера страницу регистрации |
| checkAccType() | отправляет запрос с выбором типа пользователя, который выбирает юзер при регистрации. Это влияет на вид формы авторизации. |
| clickRegistrButton( a lot of data) | отправляет данные из полей, которые заполнил пользователь контроллеру и далее сохранение нового пользователя |
| registrController | accType int  email  password  secondPassword  seekerFirstName  seekerLastName  EmployerName  EmployerDescription  workerFirstName  workerLastName  workerPatronymic  workerPhoneNumber | принимает запрос о загрузке страницы от представления и отправляет информацию о новом пользователе | getRegistrPage(accType) | в зависимости от типа аккаунта или же уровня доступа, запрашивает у модели страницу регистрации |
| setNewUserData() | отправляет данные о новом пользователе модели на обработку и сохранение в бд |
| getMainPage() | обращается к контроллеру zadazhiController с запросом отобразить главную страницу с задачами |
| registrModel | accType int  email  password  secondPassword  seekerFirstName  seekerLastName  EmployerName  EmployerDescription  workerFirstName  workerLastName  workerPatronymic  workerPhoneNumber | собирает страницы регистрации и сохраняет нового пользователя в БД | bildAMainRegistrPage() | собирает страницу с формой регистрации |
| bildEmploerRegistrPage() | собирает форму регистрации для пользователя Работодатель |
| saveNewUser(data) | сохраняет данные о новом пользователе в БД |
| avtorizView | login  password | класс описывает взаимодействие представление авторизации | showAvtorizForm() | запрашивает у контроллера страницу авторизации |
| clickEntranceButton() | запрашивает у контроллера разрешение на вход в систему |
| clickRegistrationButton() | запрашивает и переходит у контроллера переход на страницу регистрации |
| clicklFogetPassword() | запрашивает у контроллера разрешение на переход на страницу забыт пароль |
| atorizController | Login  Password | принимает запросы о авторизации от представления, отправляет и достает информацию из/в модель | getAvtorizForm() | переход на страницу авторизации |
| doEntranceAcc() | делает запрос на вход в систему |
| doRegistrationAcc() | делает запрос на переход на страницу регистрации |
| doFogetPassword() | делает запрос на переход на страницу пароль забыт |
| setEnterence() | происходит вход в систему |
| avtorizModel | Login  Password  hasAcc fool | Достает и проверяет данные об аккаунте из БД и передает их в контроллер | bildAvtorizForm()) | Собирает форму авторизации |
| correctData(login, password) | проверяет данные об аккаунте на корректность |
| MenuView | accType  imageAcc | класс описывает взаимодействие представление меню | showMenuForm() | запрашивает у контроллера страницу меню |
| clickWorkerZadachi() | отправляет запрос на переход на страницу задач работника |
| clickSettings() | отправляет запрос на переход в настройки |
| clickProfile() | отправляет запрос на переход на страницу профиля |
| clickExit() | отправляет запрос на переход на страницу выхода из системы |
| showExitForm() | запрашивает у контроллера страницу выхода |
| MenuController | accType  imgeAcc | принимает и реализует запросы от представления, отправляет и достает информацию из/в модель | getMenuForm() | запрос на переход в меню |
| setMenuForm() | делает переход в меню |
| getWorkerTasksPage() | происходит запрос на страницу задач работника |
| getSettingsPage() | запрос на переход на страницу настроек |
| getProfilePage | запрос на переход на страницу профиля аккаунта |
| getExitForm() | запрос на переход на форму выхода из системы |
| setExitForm() | делает переход на форму выхода из системы |
| MenuModel | accType  imageAcc  nameAcc | Достает данные об аккаунте из БД и передает их в контроллер | bildMenuForm(accType) | собирает форму меню аккаунта |
| getDataAcc() | получить данные аккаунта |
| ExitModel | - | Выполняет выход из системы | bildExitForm() | собирает форму выхода из системы |
| zadachiView | accType  facultyName,  educationLevel, lowestCost,  highestCost, companyName | необходим для отображения одной из основных страниц системы Задачи, а также позволяет отсортировать и отфильтровать данные страницы и выгрузить отчет | showZadachiPage() | отправляет запрос контроллеру на отрисовку страницы задачи |
| clickApplyButton (facultyName,  educationLevel, lowestCost,  highestCost, companyName) | отправляет запрос на обновление содержимого страницы исходя из выбранных пунктов фильтрации по кнопке Применить |
| clickResetButton() | отправляет запрос на фильтрации контроллеру |
| clickReportButton() | спрашивает создание отчета у контроллера |
| clickOverMenu(accType) | запрос на отрисовку Меню по клику, в зависимости от типа аккаунта пользователя |
| clickRezume() | запрос Контроллеру на переход на другую страницу Резюме |
| sortByDate()  sortByCost() | запрос на сортировку содержимого страницы |
| showReportForm() | отображает отчет по выбранным параметрам |
| zadazhiController | accType  facultyName,  educationLevel, lowestCost,  highestCost, companyName | передача запросов на отрисовку и обновление страницы между моделью и представлением и связь с reportModel | getZadachiPage()  setZadachiPage() | запрос на получение страницы Задачи от модели и отправку отрисованной страницы на представление, соответственно |
| getFiltrationApply()  getFiltrationReset() | запрос на наложение фильтрации на содержимое страницы Модели |
| getReportButton() | запрос модели reportModel на формирование отчета |
| getOverMenu(accType) | запрос на отрисовку Меню модели |
| getRezumePage() | запрос к RezumeController на отрисовку страницы Резюме |
| getSortByDate() | запрос на сортировку содержимого страницы по дате и стоимости, соответственно |
| getSortByCost() |
| setReportForm() | из reportModel возвращает форму с прорисованным отчетом Представлению |
| zadachiModel | accType  facultyName,  educationLevel, lowestCost,  highestCost, companyName  sortinType | собирается страница с опубликованными задачами, реализуется сортировка и фильтрация | bildZadachPage() | собирается страница Задачи |
| getZadachi() | вытягиваются задачи из БД |
| doFiltrationZadachi(вdata) | происходит фильтрация задач на странице по переданным параметрам |
| resetFiltrationZadachi() | происходит сброс фильтрации |
| doSorting(sortinType) | происходит сортировка задач на странице по переданным параметрам |
| reportModel | reportType, facultyName,  educationLevel, lowestCost,  highestCost, companyName,  sortType, | Достает запрашиваемые данные из БД создает на их основании отчет | bildReportForm() | строит форму страницы отчетов |
| getSortType(sortType) | запрос по типу сортировки |
| bildReport( data for a report) | строит форму страницы отчета по данным |
| resumeView | ccType  facultyName,  educationLevel | класс описывает взаимодействие представление резюме | showRezumePage() | запрашивает у контроллера страницу резюме |
| clickApplyButton (facultyName,  educationLevel) | запрашивает у контроллера разрешение на применение и сохранение данных |
| clickResetButton() | запрашивает у контроллера сброс фильтрации |
| clickReportButton() | запрашивает у контроллера отчет |
| clickOverMenu(accType) | запрашивает у контроллера меню аккаунта |
| sortByDate() | производит сортировку по дате |
| clickZadachi() | запрашивает у контроллера данные по задачам |
| showReportForm() | отобразить отчет по параметру |
| RezumeController | accType  facultyName,  educationLevel | принимает и реализует запросы от представления, отправляет и достает информацию из/в модель | getRezumePage() | запрос на переход на страницу резюме |
| getFiltrationApply(  facultyName,  educationLevel) | запрос на применение и сохранение данных |
| getFiltrationReset() | запрос на сброс фильтрации выбранных данных |
| getReportButton() | запрос на вывод отчета |
| setReportForm() | вывод отчета |
| getOverProfile(accType) | запрос на переход на профиль аккаунта |
| getSortByDate() | произвести сортировку по дате |
| getZadachiPage() | получить страницу задач |
| RezumeModel | accType  facultyName,  educationLevel | Передает данные их контроллера в модель и сохраняет их в БД | bildRezumePage() | построить страницу резюме |
| getRezume() | получить данные резюме аккаунта |
| SettingProfileView | accType, rezumeFile | класс описывает взаимодействие представление настроек | showSettingPage(accType) | запрашивает у контроллера страницу настроек |
| clickEditPasswordButton() | отправляет запрос на изменение пароля |
| clickDropAccButton() | отправляет запрос на удаление аккаунта |
| clickAddAccButton(accType) | отправляет запрос на добавление аккаунта |
| clickSaveButton() | отправляет запрос на сохранение данных |
| clickPublishRezumeButton(accType) | отправляет запрос на публикацию резюме |
| addRezumeDoc(accType,  rezumeFile) | отправляет запрос на добавление резюме |
| clickOverMenu(accType) | запрос на переход в меню аккаунта |
| getZadachiPage() | запрос на переход на страницу задач |
| getMyProfilePage() | запрос на переход на страницу профиля |
| SettingProfileController | accType | принимает и реализует запросы от представления, отправляет и достает информацию из/в модель | getSettingForm() | запрос на переход на страницу настроек |
| getEdit(AccType) | запрос на редактирование аккаунта |
| getEditPassword() | запрос на изменение пароля |
| getDropAcc() | запрос на удаление аккаунта |
| getAddAcc() | запрос на добавление аккаунта |
| getSave() | запрос на созранение данных |
| getPublishRezumeButton(accType) | запрос на публикацию резюме |
| getAddRezumeDoc(accType, rezumeFile) | запрос на добавление резюме |
| getOverMenu(accType) | запрос на переход в меню |
| getWorkersZadachiPage() | запрос на переход на страницу задач работника |
| getMyProfilePage() | запрос на переход на страницу профиля аккаунта |
| SettingProfileModel | accType | Достает данные из БД и передает их в контроллер | bildSatting(accType) | собирает форму настроек аккаунта |
| gerAccData(accType) | запрос на данные аккаунта |
| workersZadachiView | accType  facultyName,  educationLevel, lowestCost,  highestCost, companyName | отображает страницу задач опубликованных работником только для него и его Работодателя. реагирует на запрос пользователя на создание новой задачи и взаимодействие с другими страницами | showWorkersZadachiPage() | запрашивает у контроллера страницу Мои задачи |
| clickCreateButton() | запрашивает у контроллера реакцию по нажатию кнопки Создать для создания новой задачи |
| clickOnZadachaCard() | запрашивает реакцию у контроллера на клик по карточке задачи |
| ClickReportButton() | запрашивает реакцию по нажатию на кнопку Отчет |
| clickOverMenu(accType) | запрашивает реакцию у Контроллера на нажатие по Меню профиля |
| clickMyOffersPage() | запрашивает реакцию на нажатие Мои предложения |
| clickSettingPage() | запрашивает реакцию у Контроллера на нажатие по Настройки профиля |
| clickMyProfilePage() | запрашивает реакцию у Контроллера на нажатие Мой профиль |
| showReportForm() | отрисовывает форму Отчета |
| showCreateZadachaForm() | отрисовывает форму Создать задачу |
| workerZadazhiController | accType |  | getWorkerZadachiPage(accType) | передает запрос на сбор страницы Мои задачи работника модели |
| setWorkerZadachiPage() | передает страницу Мои задачи на представление |
| getCreateZadachaForm() | запрашивает у модели форму Создания задачи |
| setCreateZadachaForm() | передает форму Создания задачи представлению |
| getReportForm) | запрашивает у модели reportModel форму Отчета |
| setReportForm() | передает форму Отчета от reportModel представлению |
| getOverMenu(accType) | передает запрос на отрисовку Меню профиля MenuController |
| getMyOffersPage() | передает запрос на переход на страницу Мои предложения |
| getInfoZadachaCard() | передает запрос на получение подробной информации по клику на задачу |
| getSettingPage() | передает запрос на переход на страницу Настройки |
| getMyProfilePage() | передает запрос на переход на страницу Мой профиль |
| workerZadachiModel | accType  facultyName,  educationLevel, lowestCost,  highestCost, companyName |  | bildWorkersZadachPage() | собирает страницу Мои задачи |
| getWorkersZadachi() | запрашивает данные из БД о задачах, созданным пользователем- Работником, от которого поступил запрос |
| bildCreateZadachaForm() | собирает форму Создать задачу |
| bildInfoZadachaCard(id\_zadacha) | собирает форму с подробной информаций о задаче |

# 3. Вывод

Разработана диаграмма классов для интерфейсной и функциональной частей АИС, удовлетворяющий функционалу, описанному в ТЗ.

Дальнейшие действия: запрограммировать обозначенный в ТЗ функционал с учетом созданной БД, спроектированного интерфейса и разработанных структур классов.

# 4. Приложение

## Приложение 1. Диаграмма класса АИС “UpsilonConnect”

# 