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: **«Модуль корпоративной информационной системы сети аптек (Диаграмма классов)»**

Выполнили:

студенты гр. ЦТУ-20-2Б

Бадретдинов Эмиль Камилович

Невоструев Иван Дмитриевич

Принял:

ст. преподаватель, Банников Р.Ю.

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

*(должность, ФИО руководителя)*

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

*(оценка) (подпись)*

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

*(дата)*

**Пермь 2024**

Оглавление

[Постановка задачи 3](#_Toc159277634)

[Описание КИС и основного функционала 3](#_Toc159277635)

[Цель работы и задачи 3](#_Toc159277636)

[Диаграмма классов 4](#_Toc159277637)

[Средство разработки 4](#_Toc159277638)

[Описание диаграммы классов 5](#_Toc159277639)

[Выводы 8](#_Toc159277640)

# Постановка задачи

## Описание КИС и основного функционала

Наименование системы: «Модуль корпоративной информационной системы сети аптек».

Модуль корпоративной информационной системы сети аптек предназначен для обеспечения пользователям удобного доступа к информации по лекарственным препаратам в аптечной сети в режиме реального времени, размещения заказов, а также оптимизации управления аптеками внутри сети.

Внутри системы пользователи смогут:

1. Посматривать перечень лекарств в аптеках;
2. Совершать заказы в удобные для пользователей точки выдачи;
3. Просматривать историю заказов и их содержимое.

Администраторам системы, выступающим в роли, сотрудников аптеки будут доступны следующие функции:

1. Просмотр перечня товаров в своей аптеке;
2. Просмотр перечня заказов сделанных в аптеку с указанием основной информации по заказу описанной в ТЗ и содержимым заказа;
3. Функции по изменению статуса заказа;
4. Функции по просмотру, доступной, статистики описанной в ТЗ;
5. Функции по выгрузке отчетов из разделов, описанных в ТЗ.

Супер-пользователям системы, выступающим в роли, управляющих сети аптек, будут доступны следующие функции:

1. Добавление данных по разделам описанным в ТЗ;
2. Удаление данных по разделам описанным в ТЗ;
3. Просмотр статистики;
4. Выгрузка отчетов.

## Цель работы и задачи

Цель работы: разработать диаграмму классов для реализации модуля корпоративно-информационной системы сети аптек.

Задачи:

1. Выбрать средство реализации диаграммы классов
2. Определиться с необходимыми классами на основе ТЗ
3. Определится со связями классов и компонентов
4. Составить диаграмму классов
5. Написать описание к диаграмме

# Диаграмма классов

## Средство разработки

Диаграмма классов была разработана с помощью конструктора классов, встроенного в Visual Studio.

С помощью средства конструктора классов в Visual Studio можно создавать, визуализировать и рефакторинг схем классов в проектах C#, Visual Basic или C++. Схемы классов показывают элементы кода и их связи в области визуального конструктора, которые помогут лучше понять структуру проекта и реорганизовать код.

## Сведения о схемах классов

Вот что можно сделать с схемами классов, более подробно:

* **Разработка**: редактируйте код своего проекта путем редактирования схемы классов. Добавляйте новые и удаляйте ненужные элементы. Внесенные изменения будут отражены в коде.
* **Визуализация**: лучше понимайте структуру своего проекта путем просмотра классов в проекте на схеме классов. Настраивайте схему таким образом, что было можно сосредоточиться на наиболее важных сведениях о проекте. Сохраняйте свою схему для использования ее в дальнейшем для демонстрации или в документации.
* **Рефакторинг**: переопределяйте методы, переименовывайте идентификаторы, выполняйте рефакторинг параметров, реализуйте интерфейсы и абстрактные классы.

## Описание диаграммы классов

Диаграмма классов для Login\_Form:

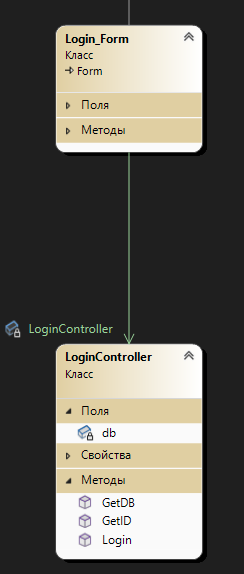


Рис. 1 – Класс Login\_Form

Диаграмма классов для Main\_Form:

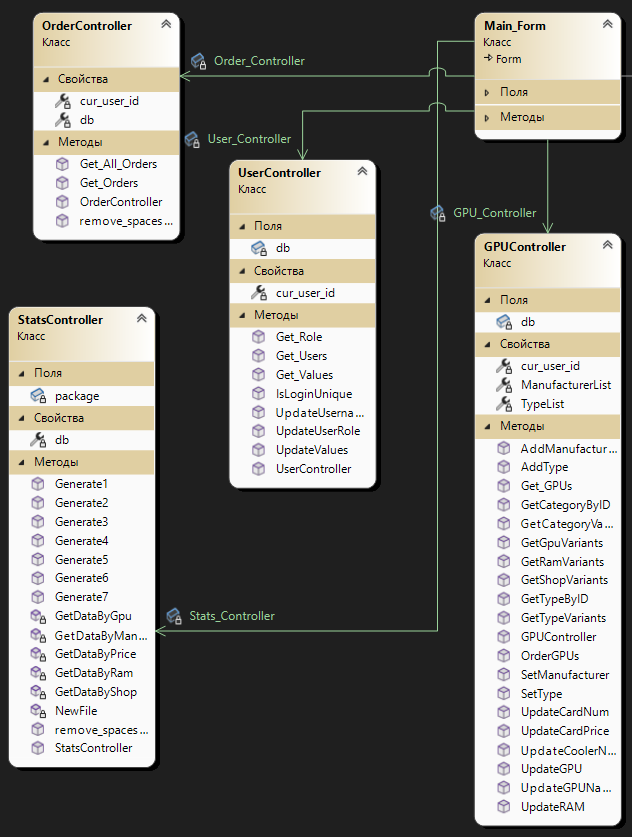


Рис. 2 – Класс Main\_Form

Диаграмма классов для Stats\_Extra\_Form:

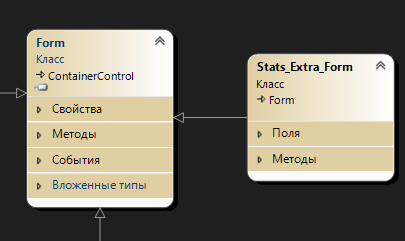


Рис. 3 – Класс Stats\_Extra\_Form

Общая диаграмма классов:

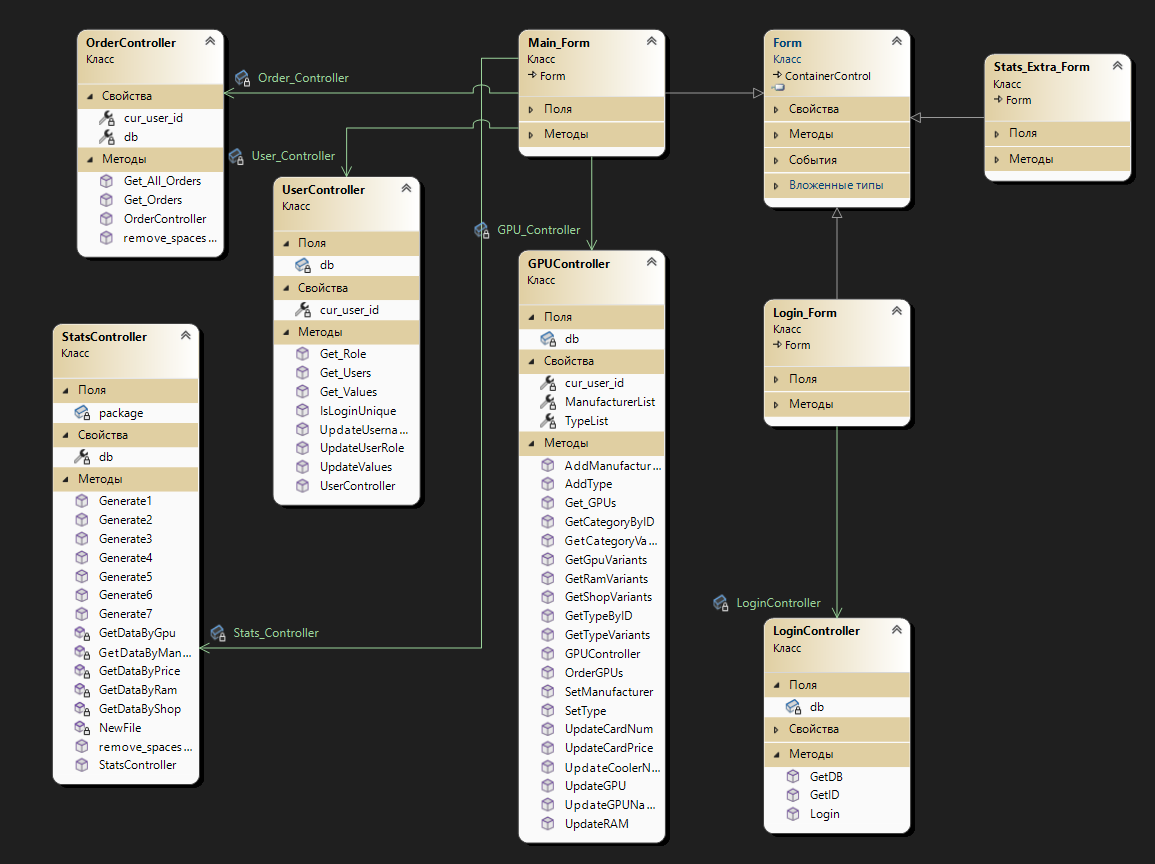


Рис. 4 – Общая диаграмма классов

## Описание классов

На диаграмме классов отображены три формы, с помощью которых пользователи взаимодействуют с системой.

**Формы**:

1. **Login\_Form** – форма авторизации пользователя. Открывается при запуске системы и отображается для всех системных ролей.
2. **Main\_Form** – главная форма, отображающая основную информацию и представляющая основной функционал для всех системных ролей.
3. **Stats\_Extra\_Form** – форма, отображающая компоненты, которые позволяют генерировать отчёты по статистике. Отображается только для пользователей с системной формой «администратор» и «модератор».

**Классы**:

1. **OrderController** – класс, обрабатывающий информацию о заказах из базы данных

Свойства:

* 1. cur\_user\_id – хранит идентификатор текущего пользователя;
  2. db – объект базы данных;

Методы:

* 1. Get\_All\_Orders – метод, выводящий информацию о всех заказах из базы данных;
  2. Get\_Orders – метод, выводящий информацию о всех заказах текущего пользователя из базы данных;
  3. OrderController – конструктор класса;
  4. remove\_spaces\_from\_table – метод, удаляющий пустые строки из двумерного массива;

1. **StatsController** – класс, генерирующий отчёты на основе информации из базы данных

Поля:

a. package – объект класса ExcelPackage, позволяющий работать с MS\_Excel;

Свойства:

* 1. db – объект базы данных;

Методы:

1. Generate(\*) – группа методов, реализующих генерацию отчётов;
2. GetDataByGpu – метод, позволяющий получить данные о видеокартах из таблицы по заданной серии графических процессоров;
3. GetDataByManufacturer – метод, позволяющий получить данные из таблицы по выбранному производителю;
4. GetDataByPrice – метод, позволяющий получить данные о видеокартах из таблицы по выбранному ценовому диапазону;
5. GetDataByRam – метод, позволяющий получить данные о видеокартах из таблицы по выбранному объему оперативной памяти;
6. GetDataByShop – метод, позволяющий получить данные о видеокартах из таблицы по выбранному магазину;
7. NewFile – метод создания нового .xls файла;
   1. remove\_spaces\_from\_ table – метод, удаляющий пустые строки из двумерного массива;
8. StatsController – конструктор класса;
9. **UserController –** класс,обрабатывающий информацию о пользователях

Поля:

* 1. db – объект базы данных;

Свойства:

* 1. cur\_user\_ id – хранит идентификатор текущего пользователя;

Методы:

1. Get\_Role – метод, позволяющий получить роль выбранного пользователя из базы данных;
2. Get\_Users – метод, позволяющий получить список пользователей;
3. Get\_Values – метод, позволяющий получить информацию о выбранном пользователе по его ID;
4. isLoginUnique – метод проверки уникальности логина пользователя;
5. UpdateUsername – метод, позволяющий редактировать имя пользователя;
6. UpdateUserRole – метод, позволяющий редактировать системную роль пользователя;
7. UpdateValues – метод, позволяющий редактировать данные о пользователе;
8. UserController – конструктор класса;
9. **GPUController –** класс, обрабатывающий информацию о видеокартах

Поля:

a. db – объект базы данных;

Свойства:

* 1. cur\_user\_id – хранит идентификатор текущего пользователя
  2. ManufacturerList – объект класса List, хранящий список всех производителей видеокарт;
  3. TypeList – объект класса List, хранящий список всех производителей графического ядра;

Методы:

1. AddManufacturer – метод, позволяющий добавить нового производителя видеокарт;
2. AddType – метод, позволяющий добавить нового производителя графического ядра;
3. Get\_GPUs – метод, позволяющий получить список всех видеокарт из базы данных;
4. GetCategoryByID – метод, позволяющий получить название производителя видеокарт по его ID из базы данных;
5. GetCategoryVariants – метод, позволяющий получить список всех производителей видеокарт из базы данных;
6. GetGpuVariants – метод, позволяющий получить список всех графических процессоров из базы данных;
7. GetRamVariants – метод, позволяющий получить список всех вариантов объёма видеопамяти из базы данных;
8. GetShopVariants – метод, позволяющий получить список всех магазинов из базы данных;
9. GetTypeByID – метод, позволяющий получить имя производителя графического ядра по его ID из базы данных;
10. GetTypeVariants – метод, позволяющий получить список всех производителей графических ядер;
11. GPUController – конструктор класса;
12. OrderGPUs – метод, позволяющий заказать видеокарту;
13. SetManufacturer – метод, позволяющий добавить нового производителя видеокарт в базу данных;
14. SetType – метод, позволяющий добавить нового производителя графических ядер в базу данных;
15. UpdateCardNum – метод, позволяющий редактировать количество видеокарт в магазине;
16. UpdateCardPrice – метод, позволяющий редактировать цену видеокарты
17. UpdateCoolerNum – метод, позволяющий редактировать количество куллеров;
18. UpdateGPU – метод, позволяющий редактировать серию графического процессора;
19. UpdateGPUName – метод, позволяющий редактировать название серии графических процессоров ;
20. UpdateRam – метод, позволяющий редактировать объем видеопамяти ;
21. **LoginController –** класс, отвечающий за авторизацию пользователей

Поля:

a. db – объект базы данных;

Свойства:

* 1. cur\_user\_id – хранит идентификатор текущего пользователя;

Методы:

1. GetDB – метод для создания объекта базы данных;
2. GetID – метод для получения ID пользователя из базы данных;
3. Login – метод, проверяющий существование пользователя с введённым логином. Сопоставляет логин с введённым паролем.

# Выводы

В ходе работы была создана диаграмма классов для системы учёта видеокарт доступных для заказа. Описаны методы и поля классов с их назначением. Согласно ТЗ следующим действием будет реализация программной части.