МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

по учебной вычислительной (ознакомительной) практике

Сласси Мутабира Ильи Каримовича

студента 1 курса, 5 группы

специальность «Прикладная математика»

Руководитель практики:

ассистент

В. Н. Романюк

Минск, 2024

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc171264827)

[Глава 1 Теоретические основы 4](#_Toc171264828)

[1.1 Общая характеристика курса «Знакомство с IT-индустрией» 4](#_Toc171264829)

[1.1.1 Модули курса «Знакомство с IT-индустрией» 4](#_Toc171264830)

[1.2 Общая характеристика курса «Контроль версий с помощью Git» 4](#_Toc171264831)

[1.2.1 Модули курса «Контроль версий с помощью Git» 5](#_Toc171264832)

[1.3 Выводы к главе 1 5](#_Toc171264833)

[Глава 2 Практическая часть курса 6](#_Toc171264834)

[2.1 Решение задач 6](#_Toc171264835)

[2.2 Лабораторная работа №1 6](#_Toc171264836)

[2.2.1 Знакомство с командной строкой операционных систем «Ubuntu», «Windows» и их основные консольные команды. 6](#_Toc171264837)

[2.2.1.1 Создание простейшего приложения на языке C++ в ОС «Ubuntu» 6](#_Toc171264838)

[2.2.1.2 Создание простейшего приложения на языке C++ в ОС «Windows» 7](#_Toc171264839)

[2.2.2 Работа с реляционной базой данных PostgreSQL 8](#_Toc171264840)

[2.3 Лабораторная работа №2 15](#_Toc171264841)

[2.4 Лабораторная работа №3 19](#_Toc171264842)

[2.5 Выводы к главе 2 20](#_Toc171264843)

[Заключение 21](#_Toc171264844)

[Список использованных источников 22](#_Toc171264845)

Введение

С 24.06 — 06.07 была пройдена учебная практика.

Целью учебной вычислительной (ознакомительной) практики является закрепление полученных знаний за первый год обучения посредством выполнения специальных учебных заданий, участия в работе над командным проектом.

Задачи практики:

* Знакомство с IT компанией, возможностями обучения и повышения квалификации, знакомство с разными направлениями профессиональной подготовки IT специалистов.
* Изучение инструментов для совместной разработки проектов и контроля версий.
* Подготовка и настройка программных сред и средств тестирования для прохождения тренинга по выбранному направлению.
* Выполнение стандартных заданий.
* Выполнение заданий повышенного уровня.
* Оформление отчета.

Ознакомительная часть:

* Экскурсия в IT компании ИУП "САМСОЛЮШНС".
* Изучение инструментов для совместной разработки проектов и контроля версий.
* Ознакомление с различными направлениями профессиональной подготовки.

Практическая часть:

* Знакомство с командной строкой «Ubuntu» и её основные консольные команды.
* Управление кодом с помощью «Git» и документирование проекта.
* Работа с реляционной базой данных «PostgreSQL».

Была проведена регистрация на следующем сайте:

- https://github.com/

ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

# 1.1 Общая характеристика курса «Знакомство с IT-индустрией»

Для ознакомления с IT-индустрией были просмотрены все образовательные лекции на платформе <https://tinyurl.com/yvt4znsb>. Каждый модуль курса был познавательным и открыл много нового.

# 1.1.1 Модули курса «Знакомство с IT-индустрией»

Ниже представлено краткое описание каждого модуля:

1. Краткий обзор IT-профессия: в видео рассказывается о различных ролях в IT-индустрии и о возможных проектах, спикер делится опытом работы и рассказывает о своей позиции в компании.
2. SAP Development & Consulting: стрим предназначен для знакомства с компанией «Цепи» и её деятельностью в области SAP, предоставляются тренинги по Java-разработке и консалтингу.
3. Data Analytics Engineering: сотрудники EPAM рассказывают о карьере в области анализа данных, объясняя, что такое аналитика данных, как она используется бизнесом и какие ценности приносит, приводятся примеры влияния аналитики данных на принятие бизнес-решений.
4. Functional Testing: беседа о профессии QA-инженера и основных аспектах тестирования программного обеспечения, описываются правила, типы и методы тестирования.
5. Robotic Process Automation: на конференции "Robotic Process Automation Junior" были представлены темы, связанные с робототехникой и автоматизацией процессов, рассматривалась технология Orpheus Robotics и её применение в автоматизированных процессах.
6. Performance Optimization: на конференции "Junior's Performance Optimization Conference" рассматривается роль и обязанности аналитика производительности, предоставляются советы для начинающих.
7. Cloud & DevOps: на выпуске Junior's DevOps Online Conference обсуждается тема DevOps и его роль на проектах, профессиональные DevOps-инженеры объясняют суть и обязанности этой роли.

# 1.2 Общая характеристика курса «Контроль версий с помощью Git»

Я зарегистрировался на сайте «GitHub» (https://github.com/) и активировал курс «Introduction to GitHub». В нём рассматриваются: создание веток, коммиты, «Pull requests», создание репозиториев, слияние веток и все фундаментальные команды «Git».

# Модули курса «Контроль версий с помощью Git»

1. Основные команды «Git»:

git --version - версия

git --config global user.name “username” - установить глобальное имя в гит git config user.name - посмотреть глобальное имя в гит

git --config global user.email “mail” - установить глобальную почту в гит

git config user.email - посмотреть глобальную почту в гит

git init - проинициализировать репозиторий

git status - посмотреть статус репозитория

git add main.cpp - отслеживать файл main.cpp в репозитории

git add \*.cpp - отслеживать все файлы в папке с расширением .cpp

git add . - добавить все не отслеживаемы файлы в локальный репозиторий

git commit -m “added new file main.cpp” - -m для того, чтобы добавить комментарий к коммиту.

1. Загрузка своего репозитория на «GitHub»:  
   git remote add origin ссылка\_на\_репозиторий – добавить удалённый репозиторий  
   git push origin master – загрузить все коммиты в главную ветку
2. Загрузка с «GitHub»:  
   git fetch origin - добавляем в наш локальный репозиторий файлы с гитхаба  
   git merge origin/master - вставляем файлы из локального репозитория в рабочую область  
   git pull - выполняет сразу 2 команды «git fetch origin» и «git merge origin/master»

# 1.3 Выводы к главе 1

Были просмотрены лекции и конференции по различным направлениям IT-индустрии, которые дали общее представление о ней.

Был пройден курс по основам контроля версий с помощью программного обеспечения «Git». Рассмотрены фундаментальные и самые важные команды «Git».

Глава 2 Практическая часть курса

# 2.1 Решение задач

Руководителем практики было предоставлено три лабораторных работы, которые необходимо было выполнить в рамках учебной практики. Все лабораторные работы выполнены.

# 2.2 Лабораторная работа №1

# 2.2.1 Знакомство с командной строкой операционных систем «Ubuntu», «Windows» и их основные консольные команды.

Целью работы было изучить основные команды работы с файлами и папками в командной строке «Windows» и в консоли ОС «Ubuntu».

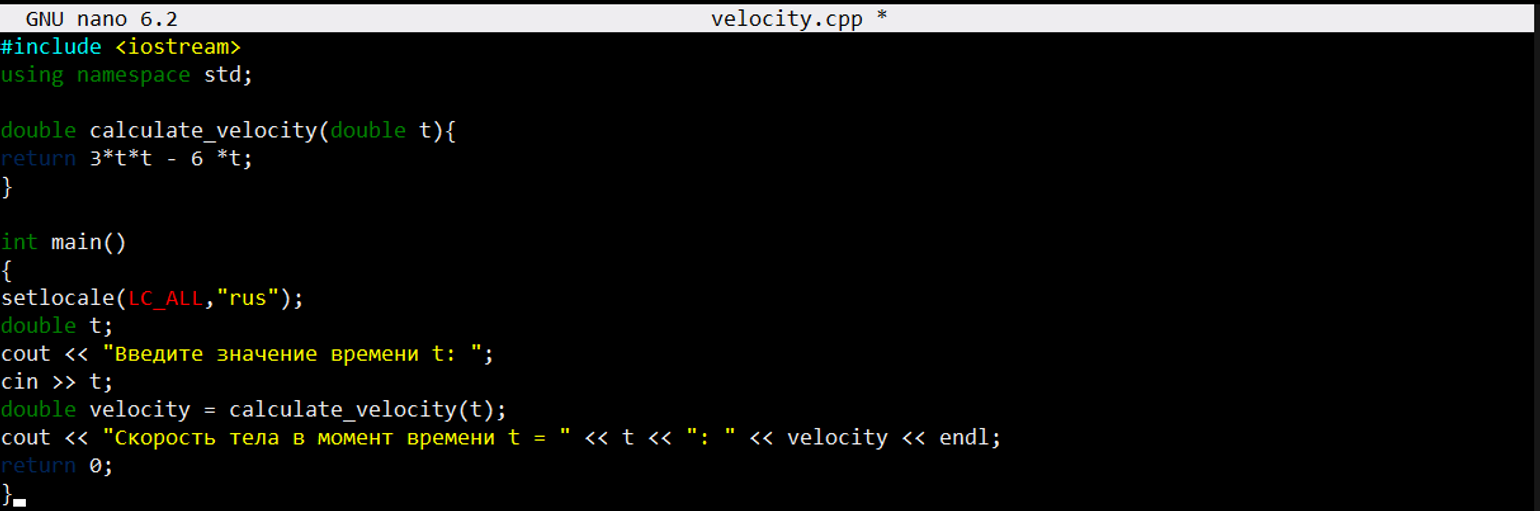
Самым первым заданием было создать простейшее приложение на языке C++ с помощью командной строки. Для этого потребовалось установить ПО «VirtualBox» – программный продукт виртуализации для операционных систем «Windows», «Linux», «FreeBSD», «macOS». На виртуальную машину можно загрузить любую операционную систему используя файлы расширения «.iso» соответствующей ОС.

# 2.2.1.1 Создание простейшего приложения на языке C++ в ОС «Ubuntu»

Я загрузил «Ubuntu» 24.04 в «VirtualBox». Задача: Тело движется по закону S = t3 - 3t2 +2. Вычислить скорость тела в момент времени t. Значение t ввести с клавиатуры (Функция скорости есть производная от функции расстояния по времени).

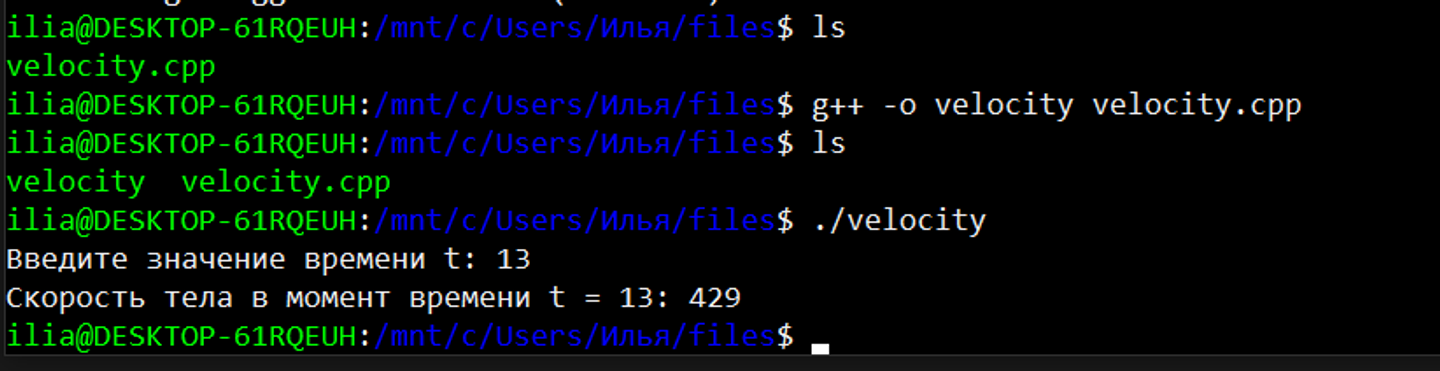
Шаги выполнения:

1. Создал файл «velocity.cpp» и зашёл в редактор кода «Nano» с помощью команды «nano velocity.cpp».
2. Написал код программы. Для сохранения изменений нажал комбинацию клавиш «Ctrl + O», затем «Enter». Для выхода из редактора кода нажал клавиши «Ctrl + X».



1. Установил компилятор «g++» с помощью команд: «sudo apt update» и «sudo apt install g++».
2. Скомпилировал файл «velocity.cpp» командой «g++ -o velocity.cpp».
3. Запустил программу командой «./velocity».

Результат выполнения программы:



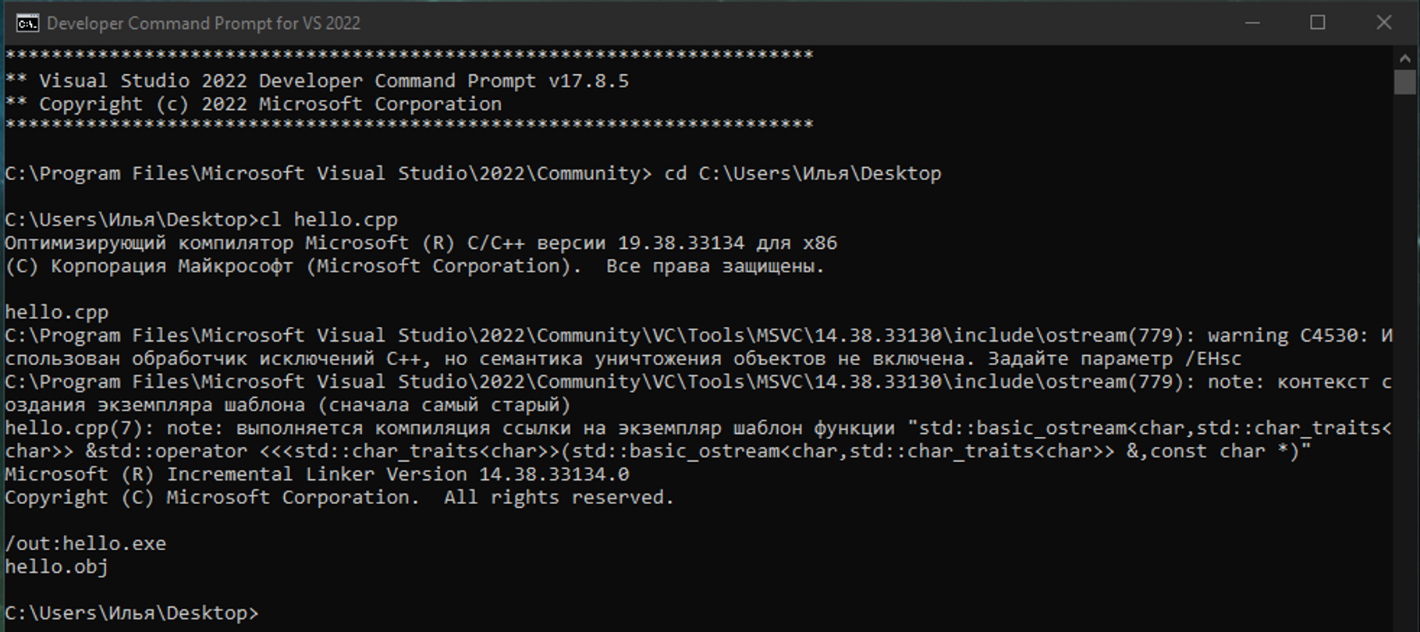
# 2.2.1.2 Создание простейшего приложения на языке C++ в ОС «Windows»

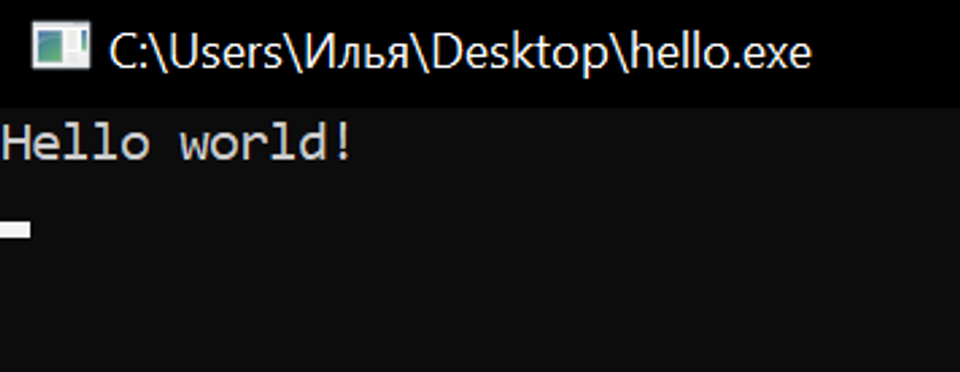
Задача: создать файл «hello.cpp», который выводит в консоль текст «Hello world!».

Шаги выполнения:

1. Создал файл «hello.cpp»
2. Нашёл в «Пуске» консоль «Developer Command Prompt for VS 2022». Эта консоль настроит необходимые переменные окружения для использования инструментов компиляции «Visual Studio»
3. Перешел в директорию, в которой лежит файл «hello.cpp»
4. С помощью команды «cl hello.cpp» скомпилировал файл.
5. Запустил файл «hello.exe»

Результат выполнения:



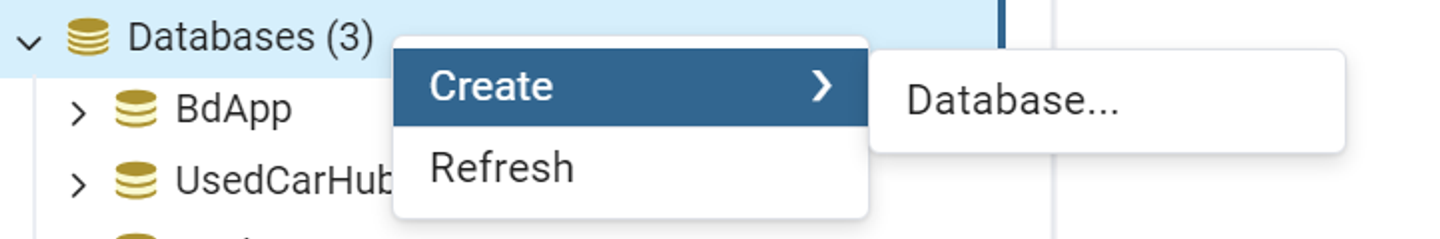
****

# 2.2.2 Работа с реляционной базой данных PostgreSQL

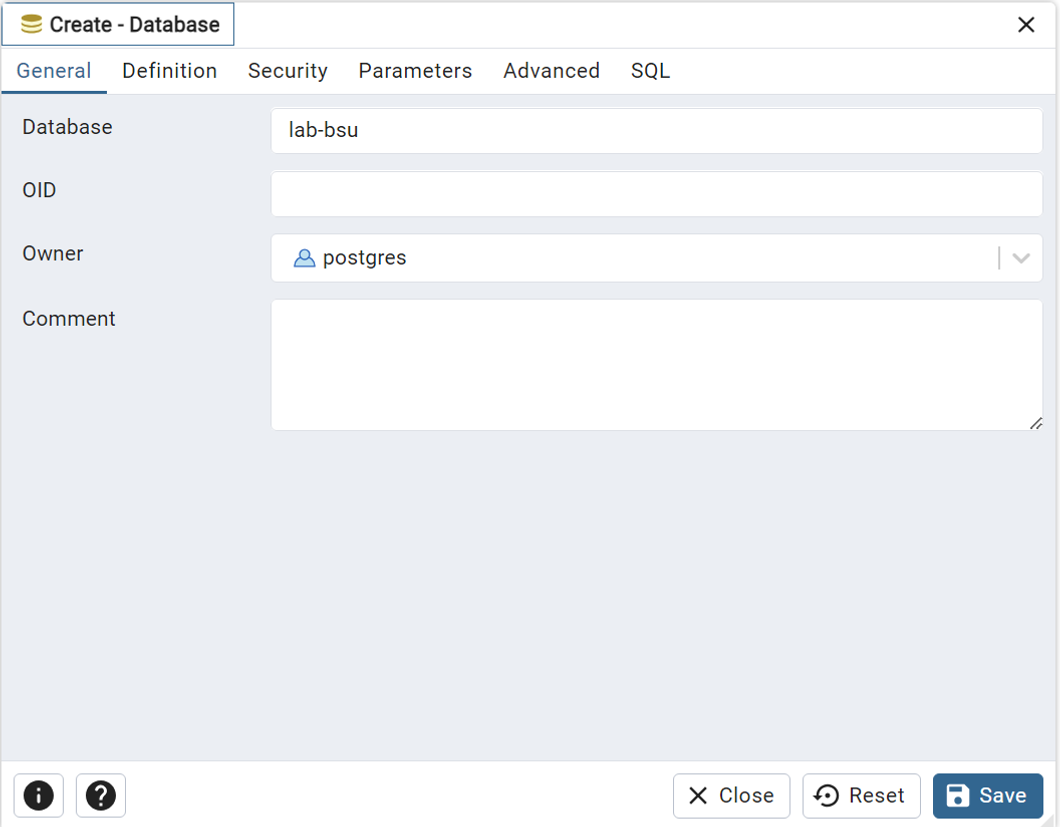
В этом задании я использовал реляционную базу данных «PostgreSQL» и приложение «PgAdmin 4» для работы с ней.

Для создания базы данных я выполнил следующие шаги:

1. Нажал на кнопку «Create», а затем «Database»



2. Дал название базе данных и нажал кнопку «Save»



**Задание**:

«Автомобиль»: id, марка, цвет, серийный номер, регистрационный номер, год выпуска, год техосмотра, цена.

Выполнить запросы:

• Вывести данные про автомобили, которым больше 5 лет.

• Создать таблицу vendor (id, ven\_name, ven\_description).

• Используя инструкцию alter, добавить дополнительные столбцы, один из

которых vendor\_id (тип integer и содержит идентификаторы заводов).

•Вывести данные обо всех автомобилях в форме идентификатор автомобиля, наименование, год выпуска, название завода.

• Подсчет количества машин с помощью count, если стоимость>16000 руб

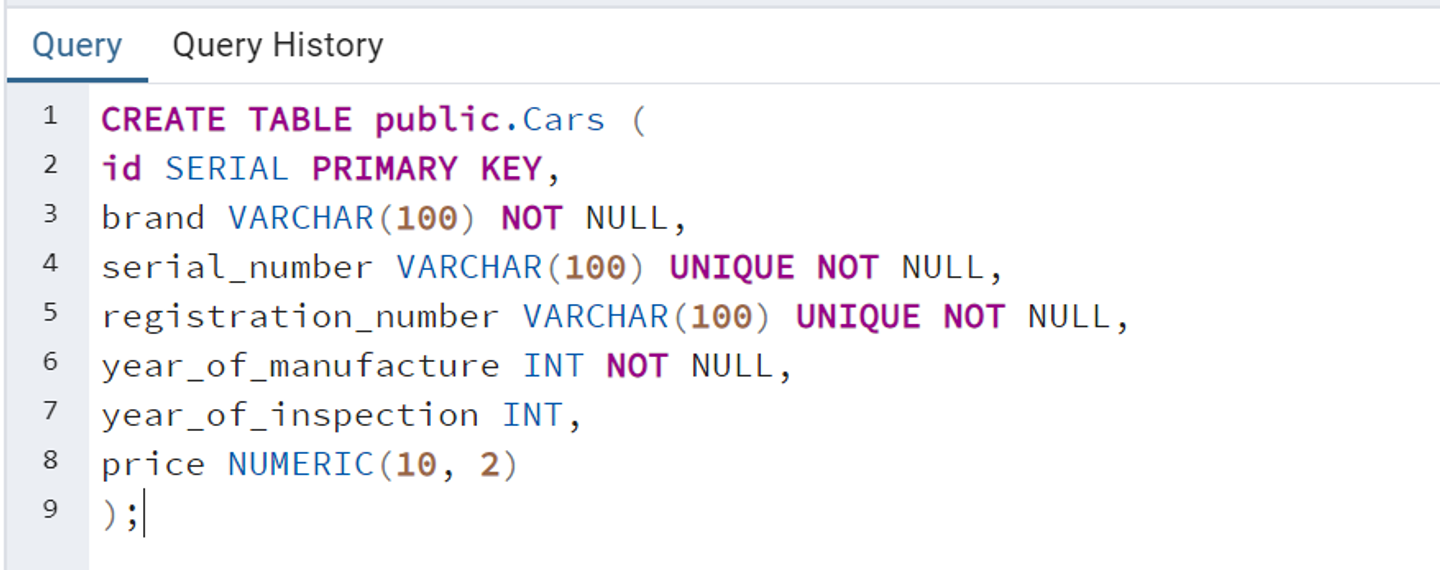
• Суммарная стоимость машин с помощью sum, если год выпуска=2016

• Максимальная и минимальная стоимость с помощью max и min

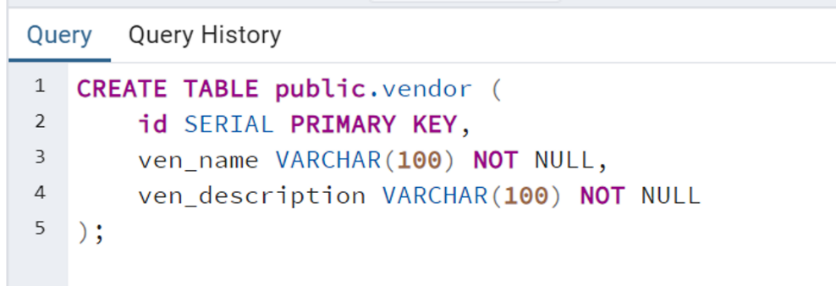
• Используя инструкцию inner join вывести полные сведения о машинах и заводе для завода с id=3.

Вот по шагам как я делал данное задание:

1. Создал таблицу для «Автомобиля» с помощью следующего SQL запроса:

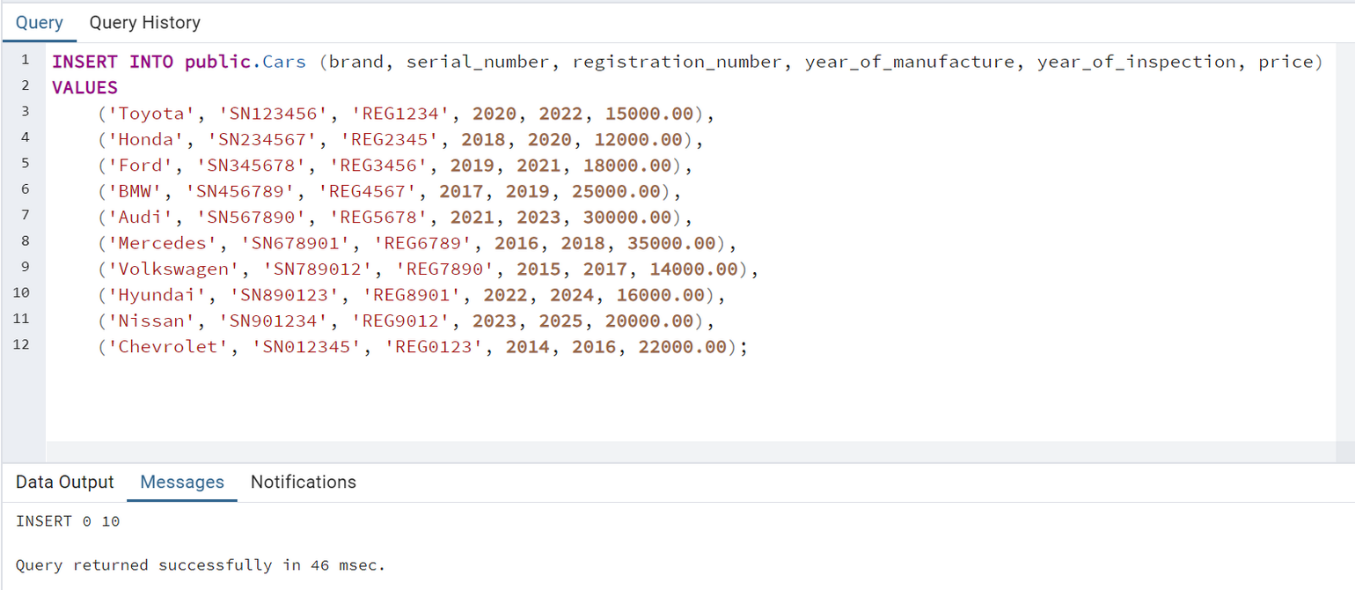


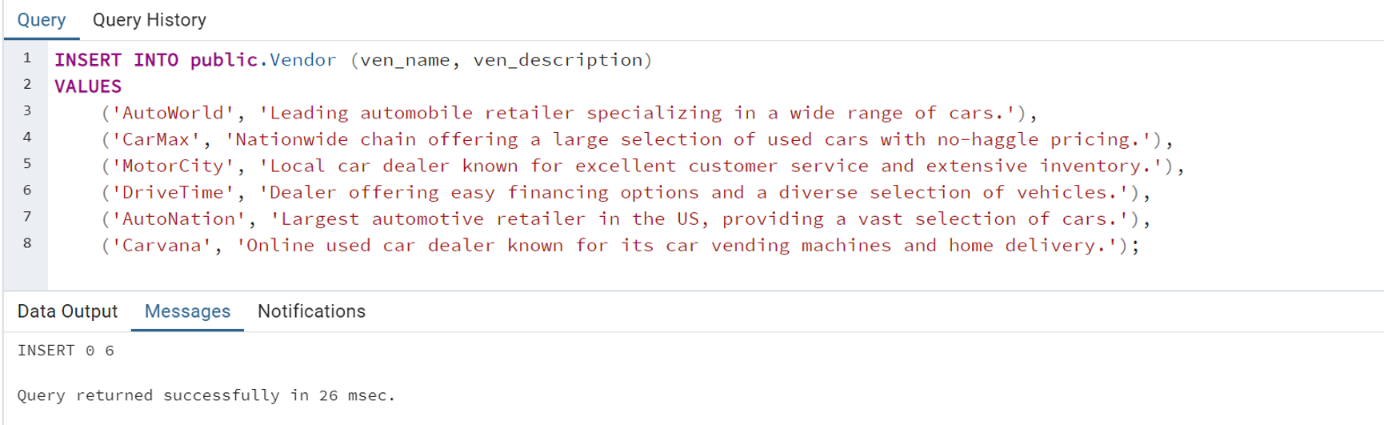
1. Создал таблицу «Vendor» с помощью следующего SQL запроса:



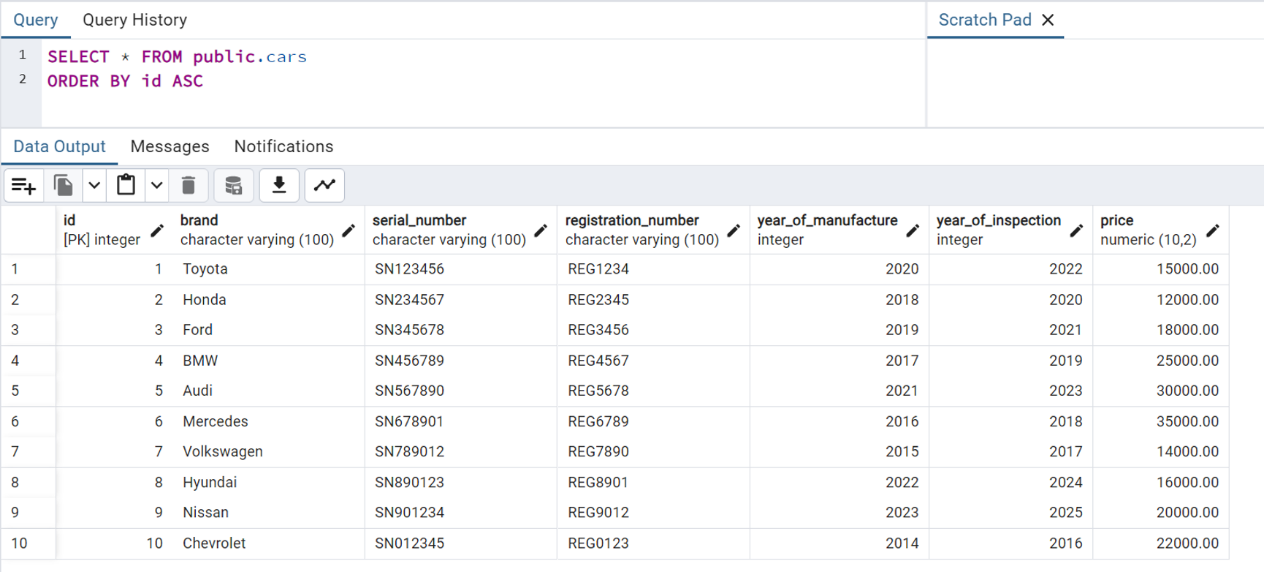
В результате появятся две таблицы

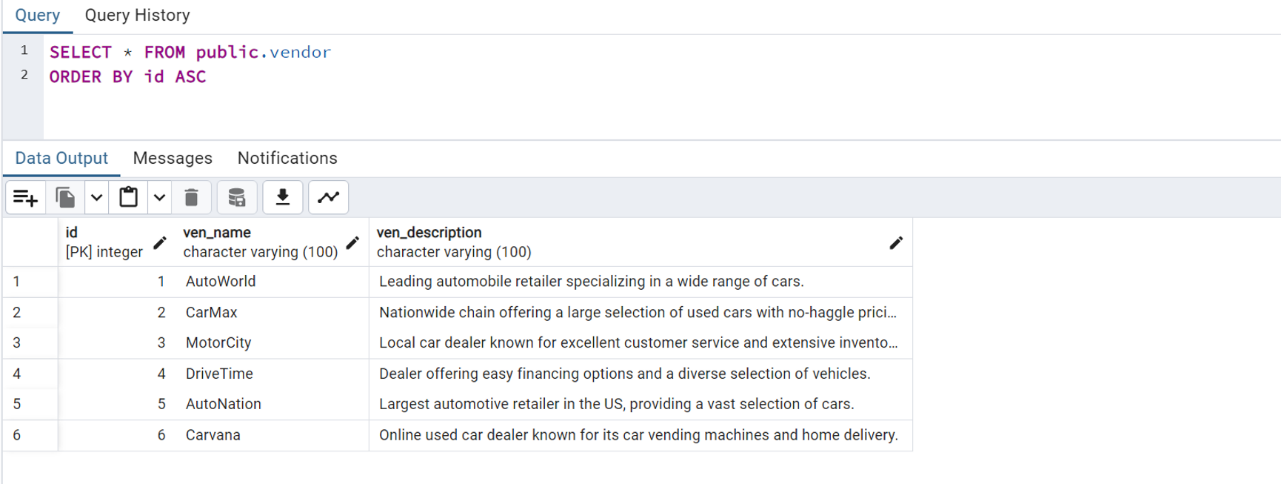


1. Для добавления данных в таблицу «Cars» выполнил следующий SQL запрос: 
2. Для добавления данных в таблицу «Vendor» требуется выполнить следующий SQL запрос:

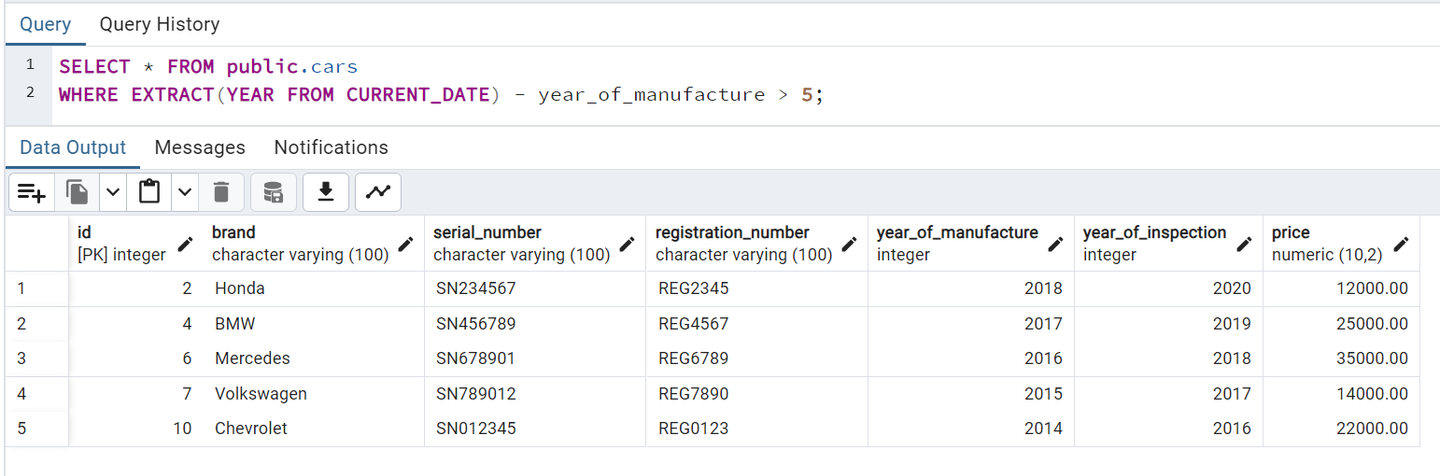


Проверил, что все данные успешно добавлены с помощью следующих SQL запросов:

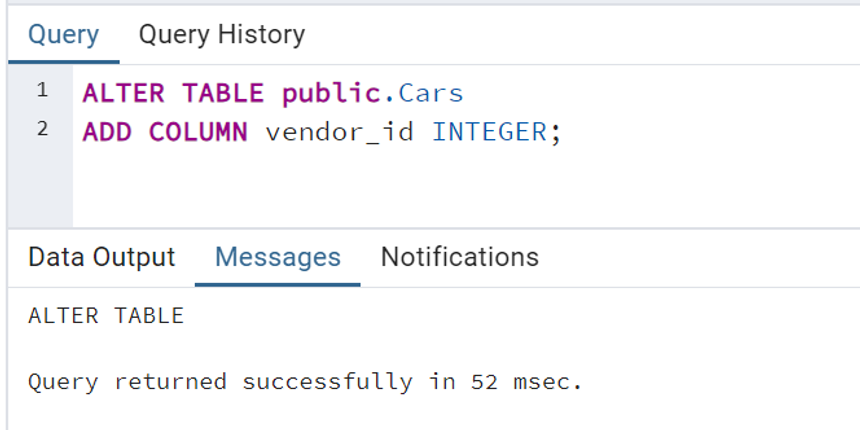




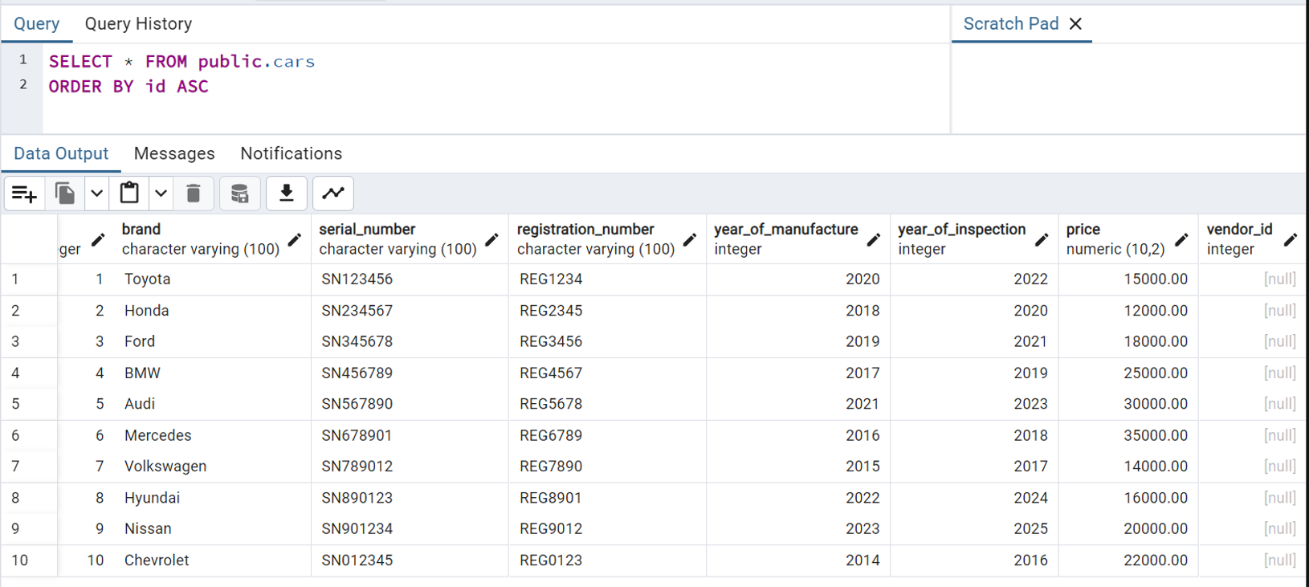
1. Вывел данные про автомобили, которым больше пяти лет:



1. Используя конструкцию «alter», добавил дополнительный столбец в таблицу «Cars», который содержит идентификаторы заводов:



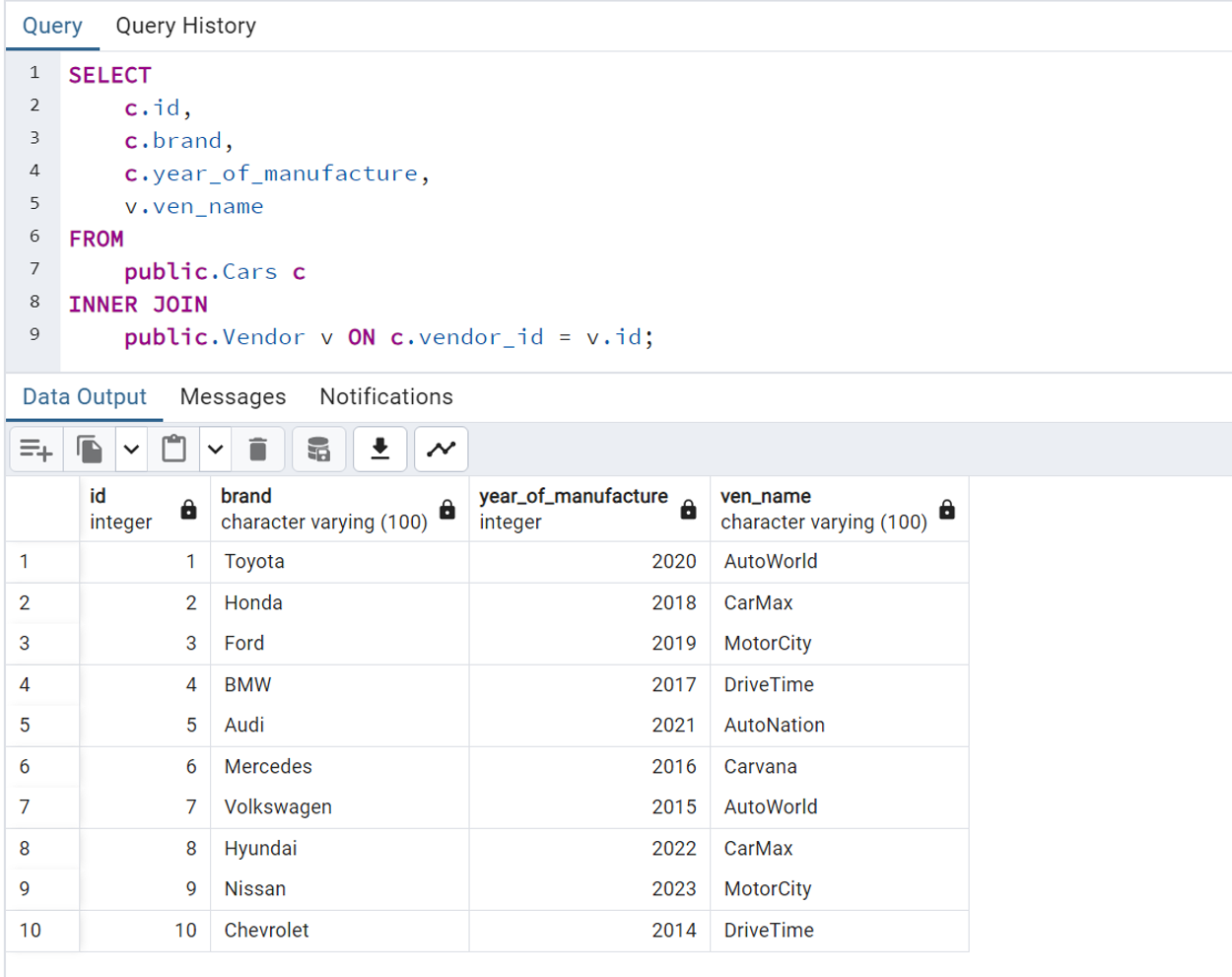
Результат добавления столбца:



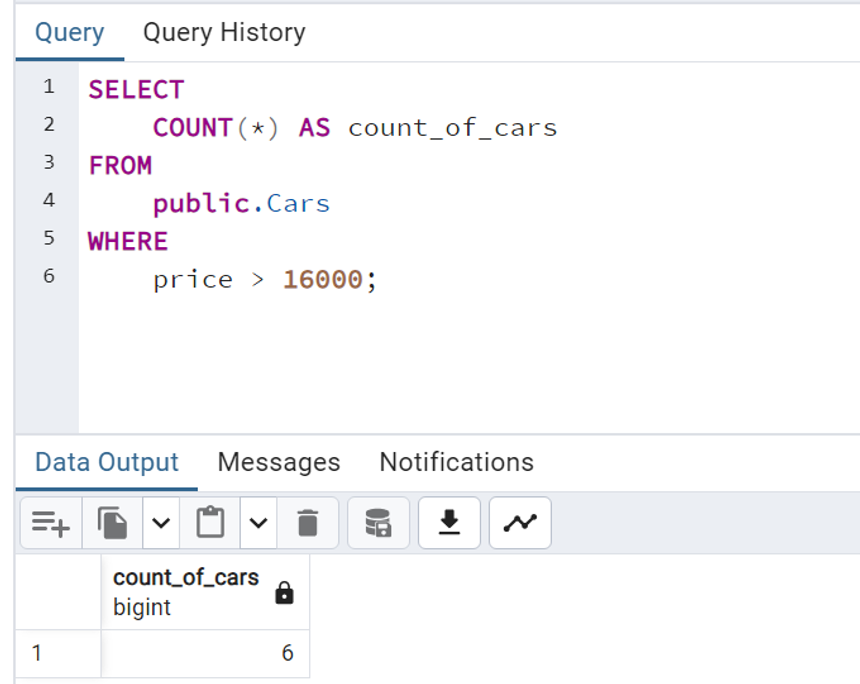
1. Заполнил колонку «vendor\_id» и вывел данные обо всех автомобилях в форме: идентификатор автомобиля, наименование, год выпуска, название завода.

|  |  |
| --- | --- |
|  |  |

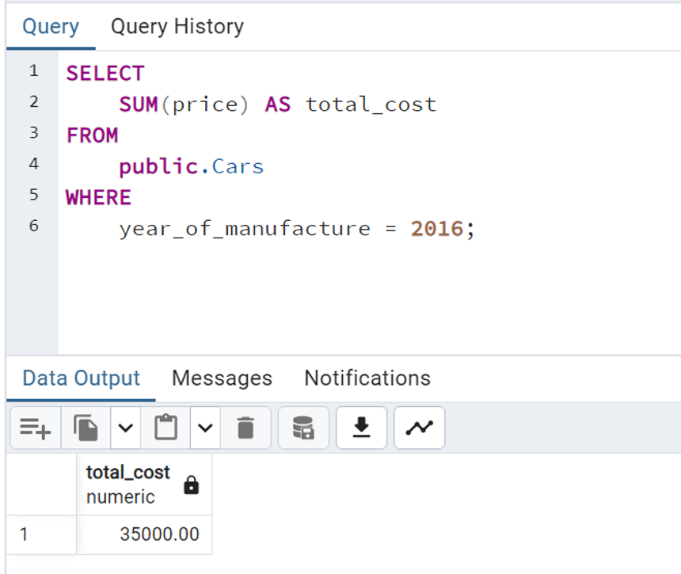
Выполнил SQL запрос для получения таблицы в нужном виде:



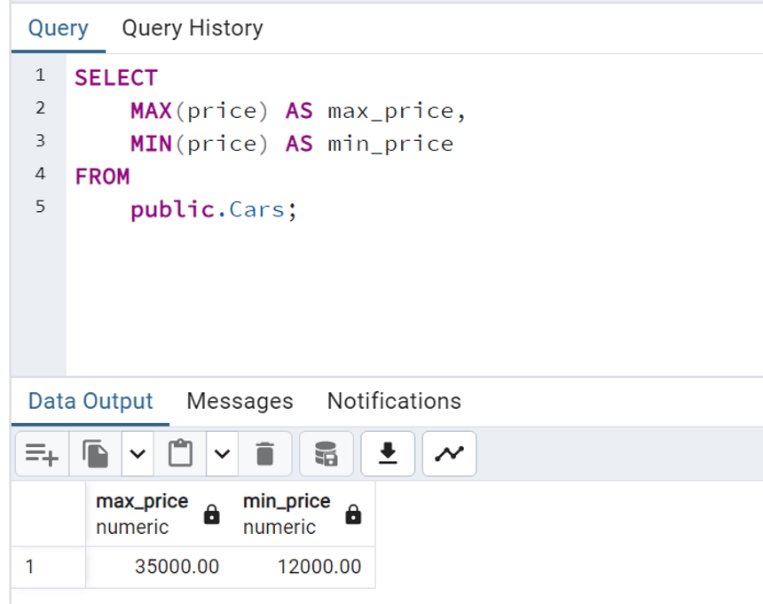
1. Подсчитал количество машин с помощью «count», если стоимость больше 16000



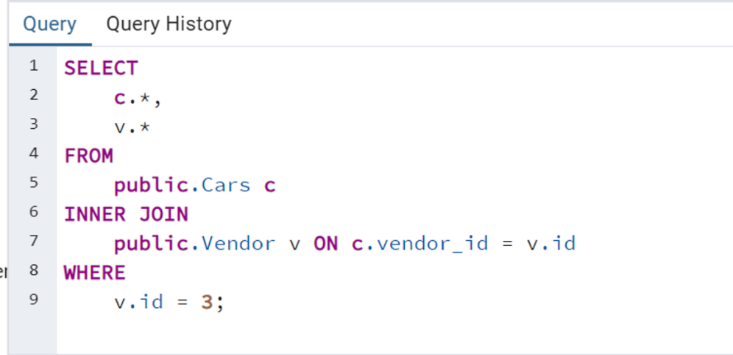
1. Вывел суммарную стоимость машин с помощью «sum», если год выпуска 2016

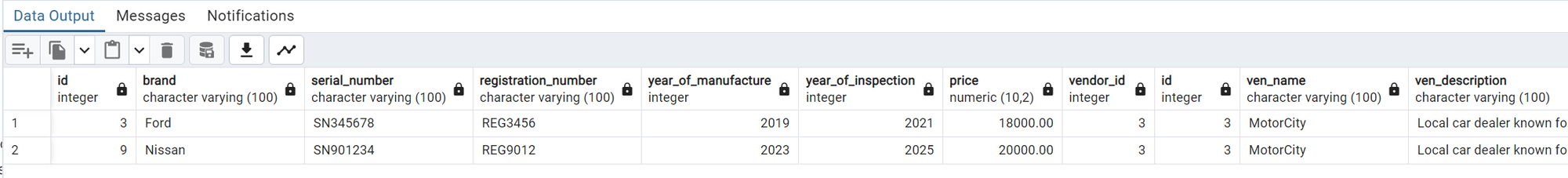


1. Вывел максимальную и минимальную стоимость машин с помощью «max» и «min»



1. Используя конструкцию «inner join» вывел полные сведения о машинах и заводе для завода с id=3



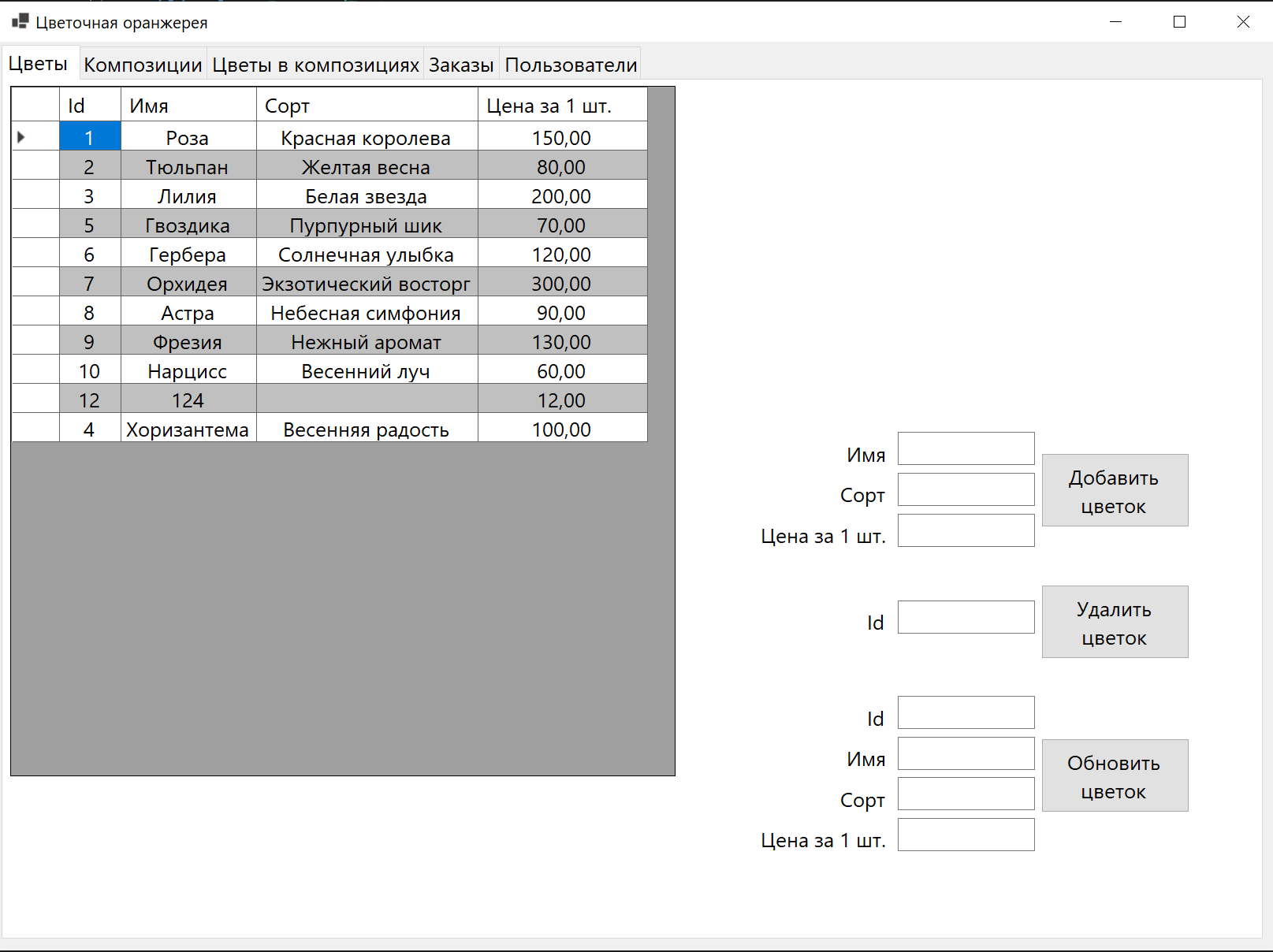
****

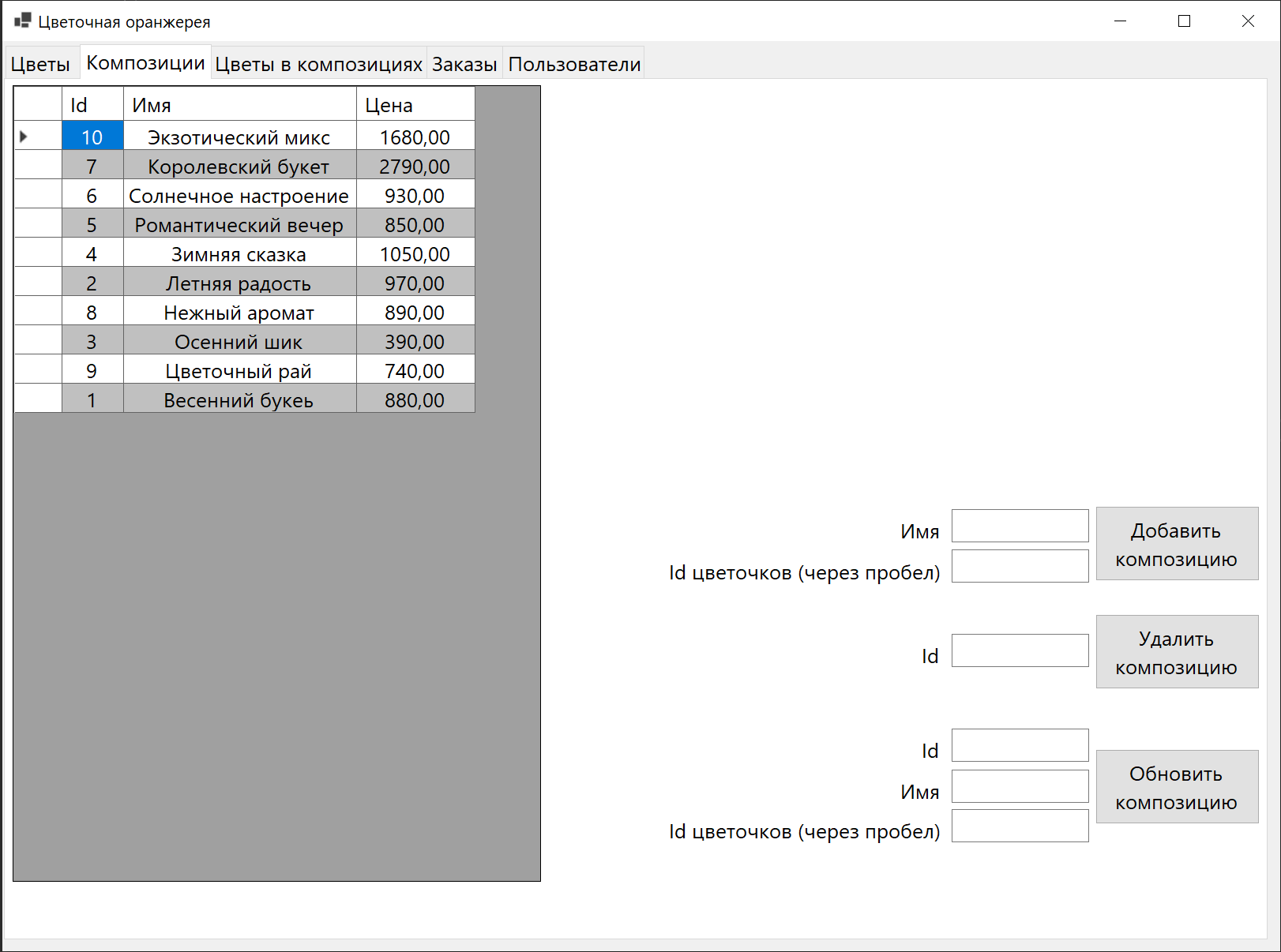
# 2.3 Лабораторная работа №2

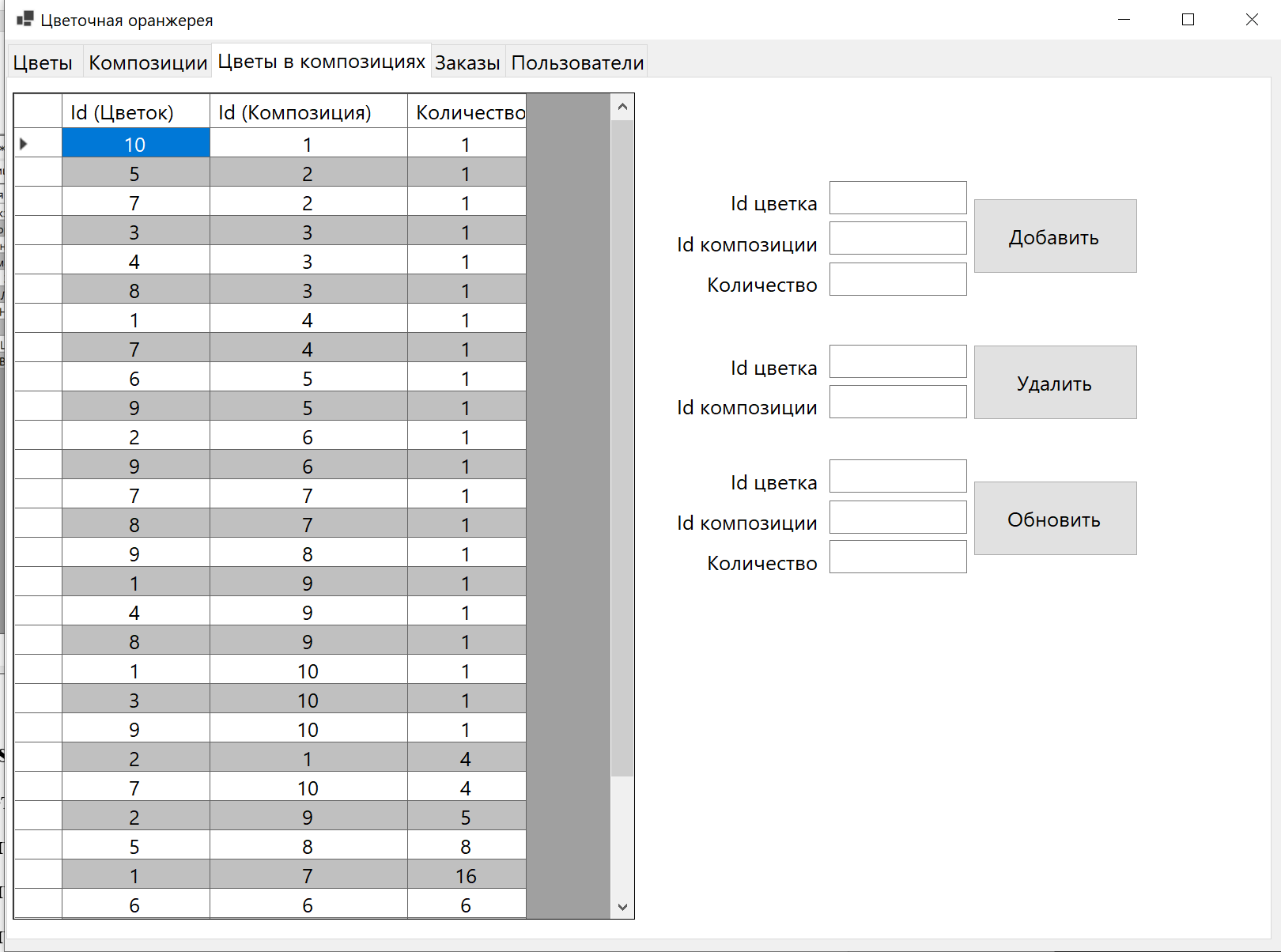
**Создание оконного приложения с использованием базы данных «PostgreSQL».**

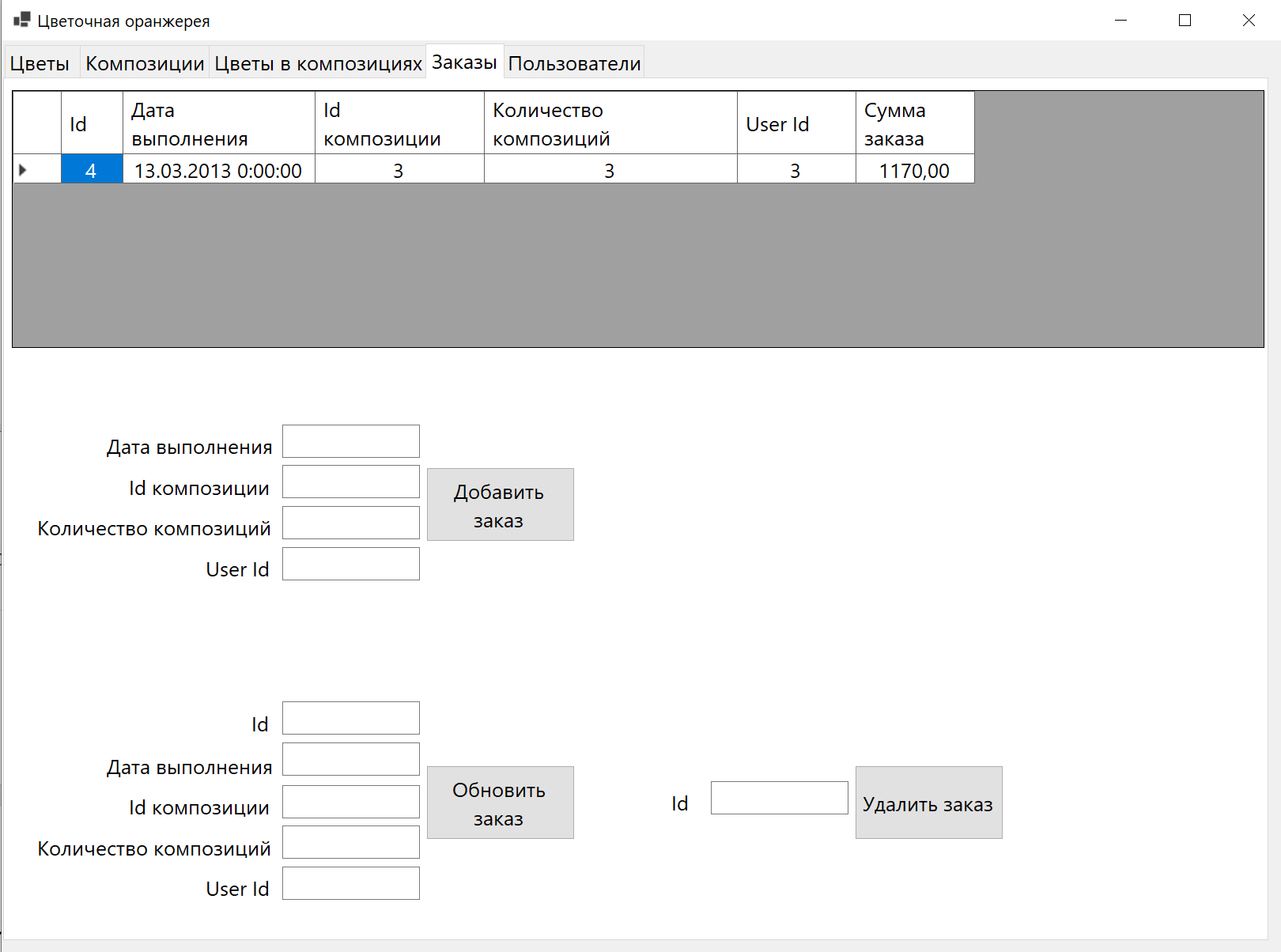
Сутью задания было создать оконное приложение с использованием базы данных. Язык программирования и технологии можно было выбрать самостоятельно. Я выбрал язык «C#», технологию «WinForm», «PostgreSQL» и «EntityFrameworkCore» для более удобной работы с базой данных с помощью языка «C#». Приложение называется «Цветочная оранжерея». В приложении нужно было написать функционал для добавления, удаления и обновления сущности «Цветок» в базе данных. Также из цветков можно создавать композиции. Композиция может содержать несколько цветков одного или разных сортов и название. Дополнительно создана таблица для реализации связи «многие ко многим» цветка и композиций. Композиции также можно добавлять, удалять, обновлять. Дополнительно можно изменять количество цветов с конкретным названием и сортом в конкретной композиции. Также в базе данных должна быть таблица «Users» для хранения логинов и паролей пользователей и таблица «Orders» для отображения истории заказов, в которой хранится информация о том, когда выполнили заказ, что в него входило, какая его стоимость, какой пользователь сделал этот заказ. Требовалось написать «валидатор» для пароля и логина пользователя. Проверить работу «валидатора» требовалось с помощью «Unit тестирования» и написания этих тестов. Полный исходный код и демонстрация работы приложения доступны по ссылке в репозитории «GitHub»: <https://github.com/smik91/BSU_summer> в директории «Лабораторная 2».

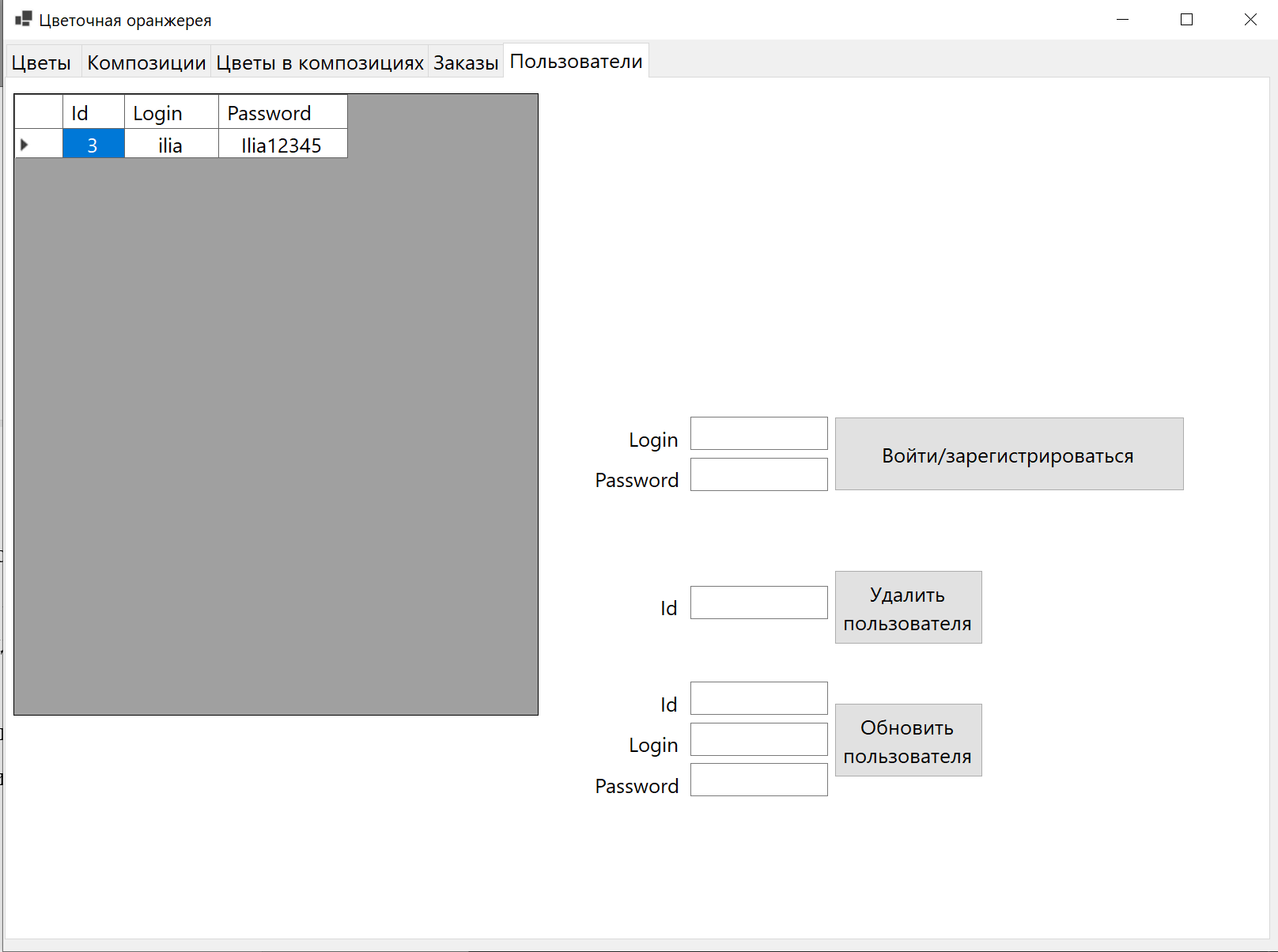
Подробнее про функционал: при удалении или обновлении цветка, также обновляется стоимость композиций, в которые он входит и их состав. При обновлении таблицы для связи цветков и композиций, также обновляется стоимость и состав композиции. Но при обновлении цветка или композиции не обновляется стоимость заказа. Вот несколько «скриншотов» работающей программы:

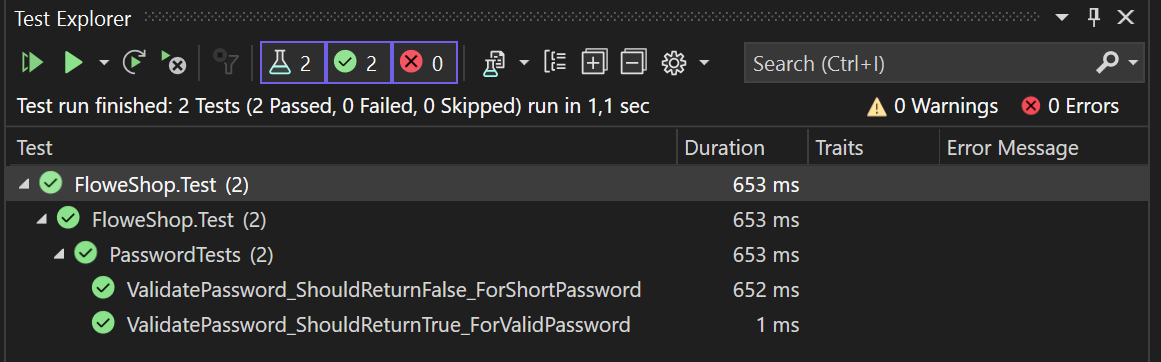






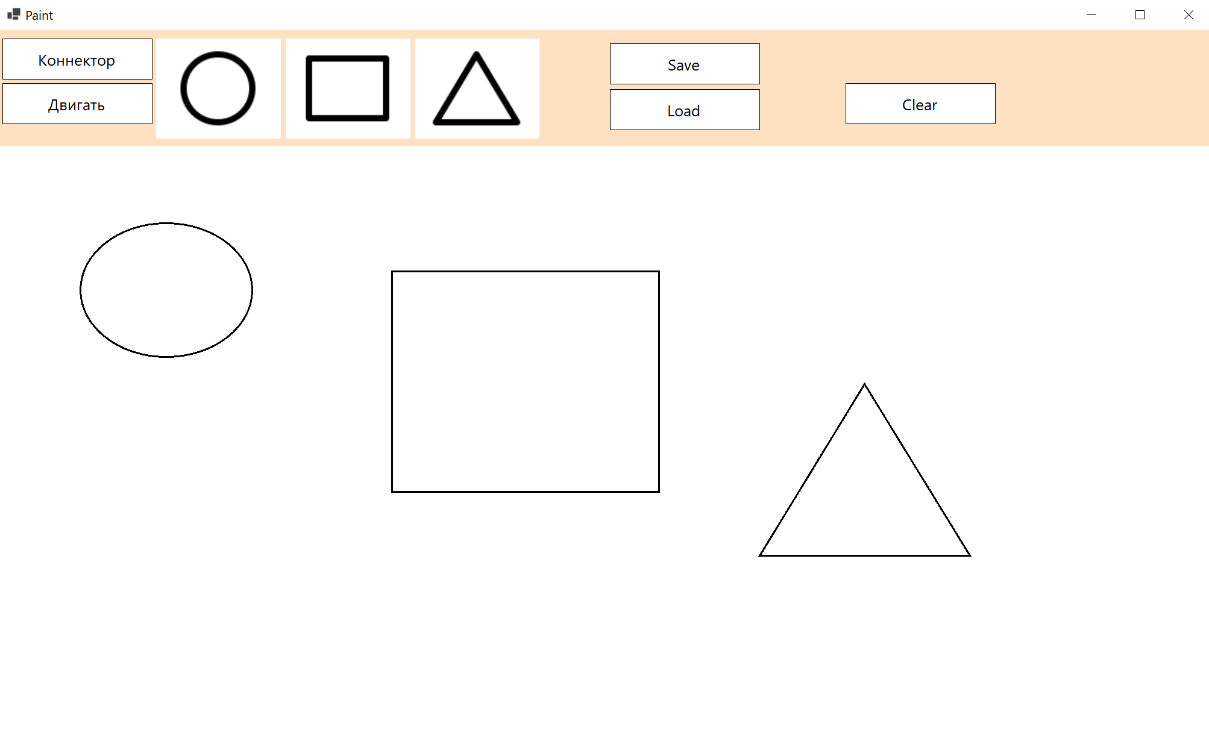


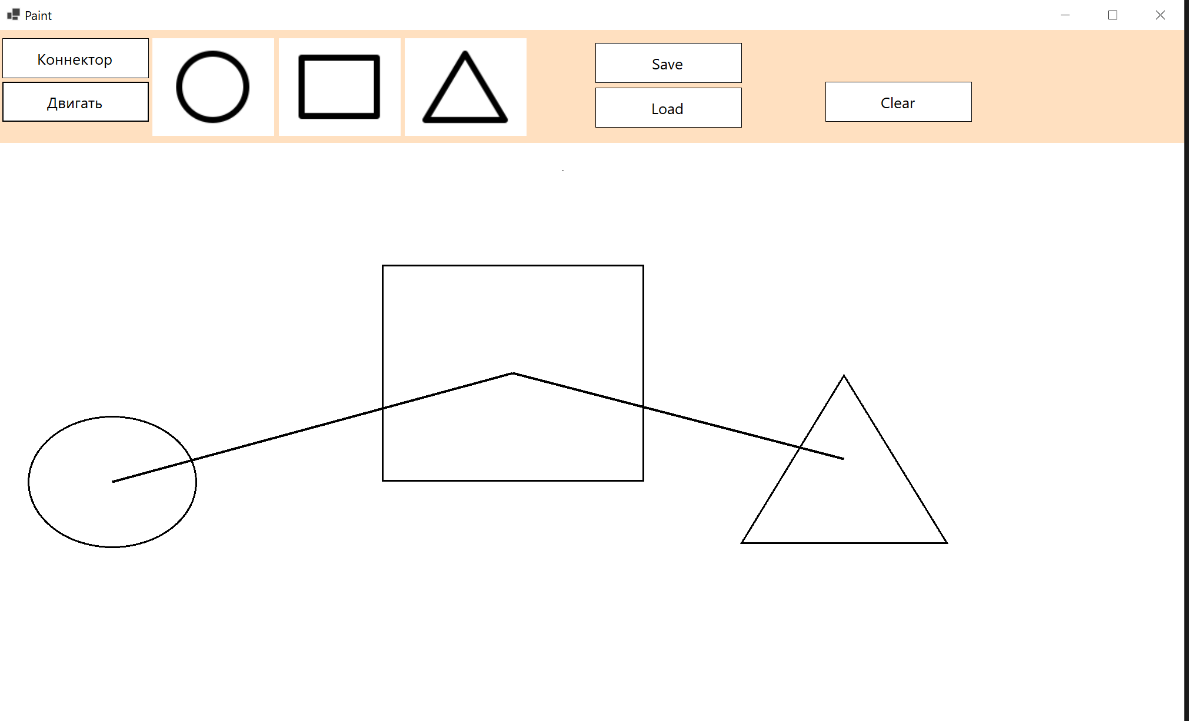


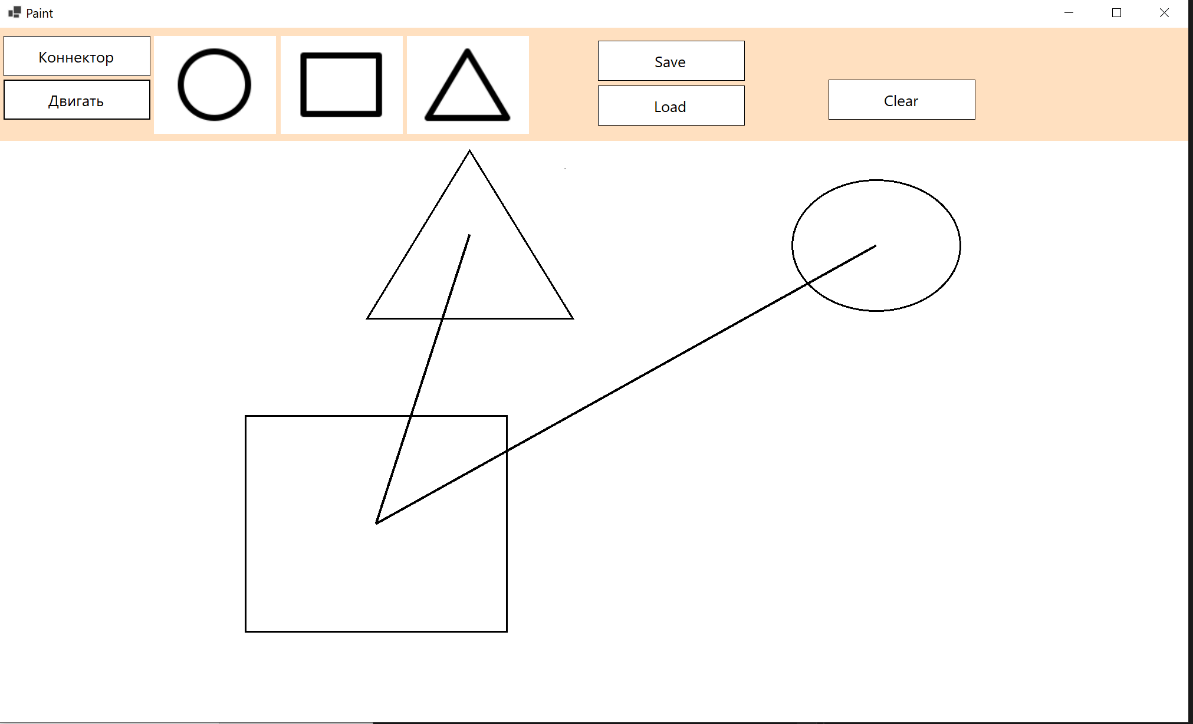


# 2.4 Лабораторная работа №3

Сутью задания было создать оконное приложение подобие приложения «Paint» с очень скромным функционалом. Главной задачей являлась реализация соединения фигур с помощью «коннектора». Фигуры для рисования выбирал пользователь среди: прямоугольника, треугольника и эллипса произвольного размера. Также требовалось реализовать перемещение фигур нажатием мыши, очистка полотна, сохранение полотна таким образом, чтобы в будущем можно было снова загрузить полотно в приложение и двигать фигуры, создавать связи и т.д. «Коннектор» это прямая линия от центра одной фигуры до центра другой фигуры. Фигуры выбираются пользователем. При перемещении фигур также изменяется и «коннектор», при этом связывая те же фигуры. Для выполнения задания язык программирования и технологии можно было выбрать самостоятельно. Я выбрал язык «C#» и технологию «WinForm» для создания оконного приложения. Весь исходный код и демонстрация работы приложения доступны по ссылке в репозитории на «GitHub»: <https://github.com/smik91/BSU_summer> в директории «Лабораторная 3». Несколько «скриншотов» работающей программы:







# 2.5 Выводы к главе 2

В результате выполнения трёх лабораторных работ были рассмотрены следующие темы: использование реляционных баз данных на основе SQL, выполнение SQL запросов к базам данных, создание оконных приложений, интегрирующих с реляционной базой данных, создание оконного приложения с использованием графики и рисовки компонентов в реальном времени.

Заключение

В результате прохождения учебной практики я ознакомился с порталом компании, возможностями обучения и повышения квалификации, с различными направлениями профессиональной подготовки специалистов в IT. Я изучил систему контроля версий «Git», научился пользоваться сайтом «GitHub», узнал, что такое реляционные базы данных. Успешно закрепил теоретические знания на практике, приобрёл ценный опыт и широкий спектр знаний о системе контроля версий «Git», базе данных «PostgreSQL» и технологии создания оконных приложений «WinForm».

Список использованных источников

1. 56 полезных команд терминала в macOS на все случаи жизни [Электронный ресурс]. – Режим доступа: <https://lifehacker.ru/komandy-terminala-macos/> – Дата доступа: 06.06.2024.
2. Github Git Guide [Электронный ресурс]. – Режим доступа: <https://github.com/git-guides> – Дата доступа: 06.06.2024.
3. Mac Terminal cheatsheet [Электронный ресурс]. – Режим доступа: <https://zhuosongz.github.io/docs/cheatsheets/mac_terminal.pdf> – Дата доступа: 06.06.2024.
4. METANIT.COM Сайт о программировании [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/tutorial/1.2.php. – Дата доступа: 06.06.2024.
5. Брайен, В., Деннис, М. Язык программирования C [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/97/97/info> – Дата доступа: 06.06.2024.
6. Дейтел, Х., Дейтел, П. Как программировать на C.
7. Документация по C# [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/dotnet/csharp/ – Дата доступа: 06.06.2024.
8. Иванов, Н.Н. Программирование в Linux. Самоучитель. – СПб.: БХВ-Петербург, 2007. – 416 с.
9. Калинина, Н.А., Костюкова, Н.И. Основы программирования на языке C [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/43/43/info> – Дата доступа: 06.06.2024.
10. Керниган, Б., Ритчи, Д. Язык программирования C. Второе издание. – М.: Вильямс, 2008. – 304 с.
11. Консоль для маководов: Beyond the GUI [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/143341/> – Дата доступа: 06.06.2024.
12. Начинаем работать с VMware Workstation (для чайников) [Электронный ресурс]. – Режим доступа: <http://all-ht.ru/inf/vpc/p_0_2.html> – Дата доступа: 06.06.2024.
13. Начинающим о работе в терминале [Электронный ресурс]. – Режим доступа: <http://appleinsider.ru/tips-tricks/mac-os-xnachinayushhim-o-rabote-v-terminale.html> – Дата доступа: 06.06.2024.
14. Основы работы с командной строкой (Терминалом) на Mac OS X [Электронный ресурс]. – Режим доступа: <https://vertex-academy.com/tutorials/ru/komandy-komandnaya-stroka-terminal-vmacos/> – Дата доступа: 06.06.2024.
15. Редакторы nano и vim [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=R33F0EDivwk> – Дата доступа: 06.06.2024.
16. Руководство по PostgeSQL [Электронный ресурс]. – Режим доступа: <https://metanit.com/sql/postgresql> – Дата доступа: 06.06.2024.
17. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5: основные принципы и практики программирования / Эндрю Троелсен, Филипп Джепикс. – Киев: Диалектика, 2022. – 772 с [1 том], 634 с [2 том]. (Книга с двумя авторами)
18. Хабр Q&A – Вопросы и ответы для IT-специалистов [Электронный ресурс]. – Режим доступа: https://qna.habr.com/q/938443 – Дата доступа: 06.06.2024.