

**Частное учреждение профессиональная образовательная организация  
«СТОЛИЧНЫЙ БИЗНЕС КОЛЛЕДЖ»**

**ЛАБОРАТОРНАЯ РАБОТА**

Подключение к базе данных MySQL в консольном приложении при помощи языка программирования C# и библиотеки MySql.Data в интегрированной среде разработки Visual Studio 2022.

специальность 09.02.03 Программирование в компьютерных системах

Выполнил студент:

\_\_\_\_\_

Д. Д. Суравенков

(подпись)

Группы 1404ПК.

Руководитель :

\_\_\_\_\_

(подпись)

Л.А. Киселёв

**Москва**

2022

1. Запустить Visual studio 2022.

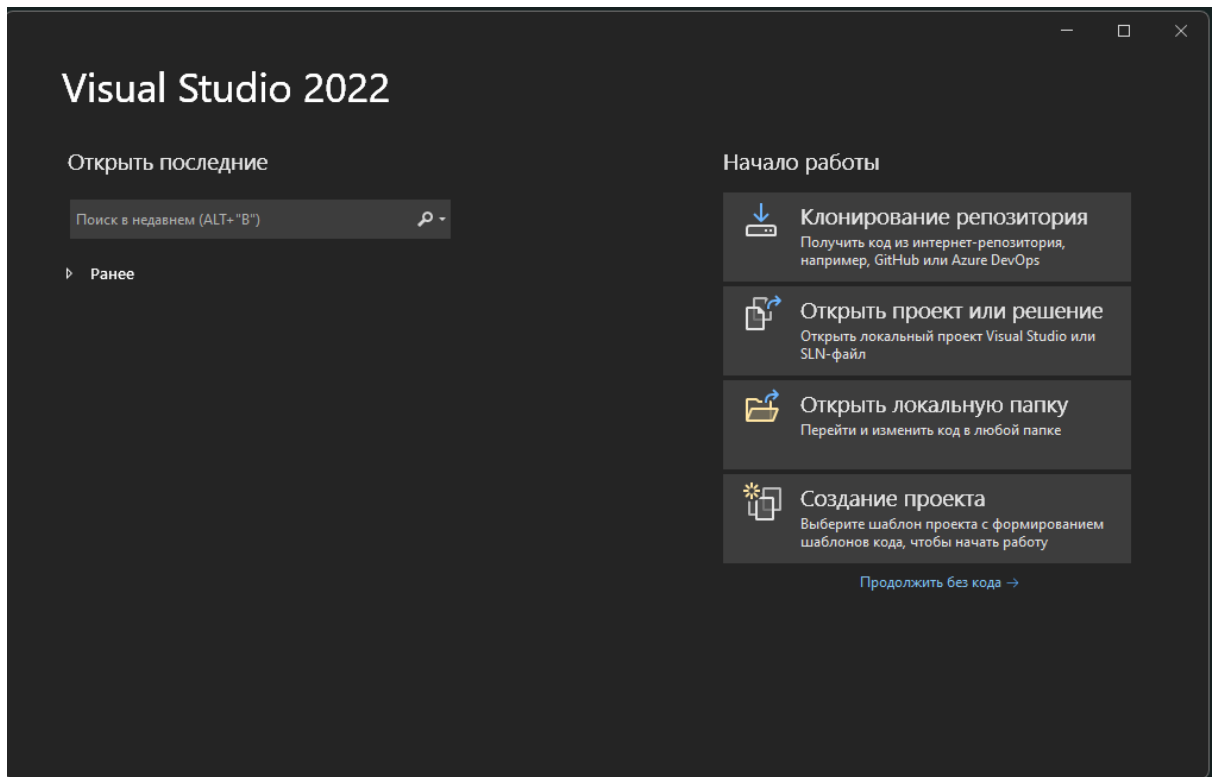


Рисунок 1. Приветственный экран Visual Studio 2022.

2. Нажать на кнопку “Создание проекта”.

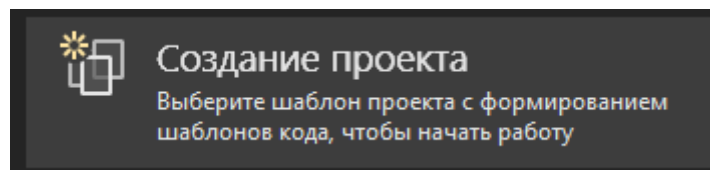


Рисунок 2. Кнопка создания проекта.

3. Откроется окно выбора шаблонов для проекта.

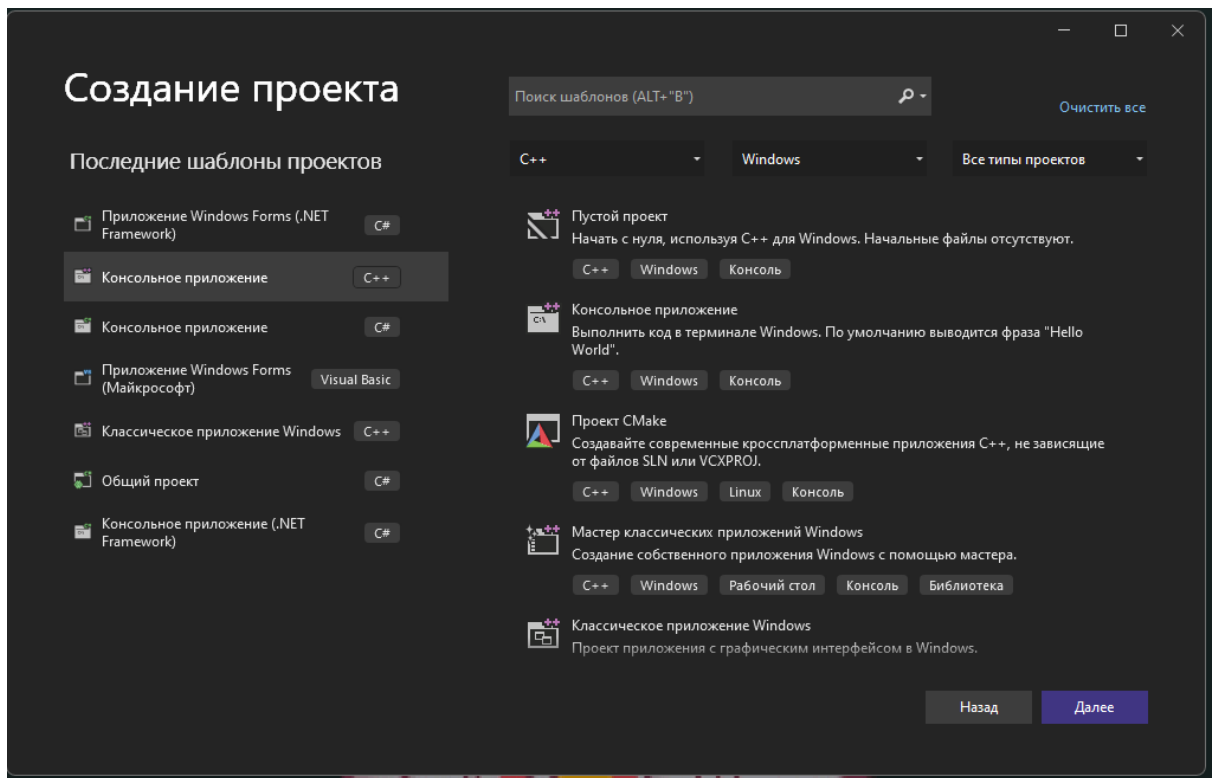


Рисунок 3. Окно выбора шаблона проекта.

4. В списке фильтров указать язык программирования C#, операционную систему Windows и типа проекта оставить “Все типы проектов”.

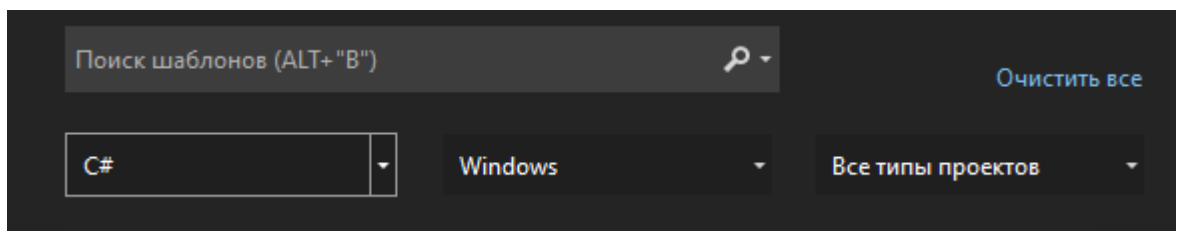


Рисунок 4. Поиск и фильтры отображение шаблонов.

5. В списке шаблонов выбрать “Консольное приложение”.

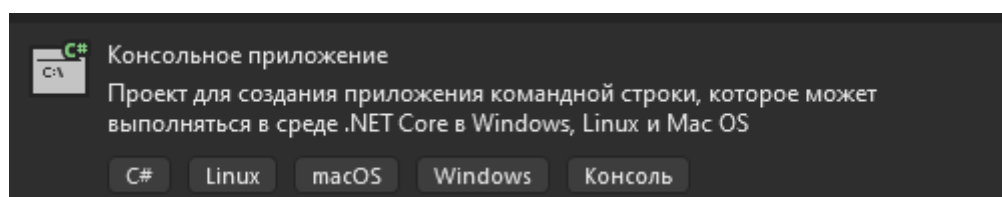


Рисунок 5. Шаблон “Консольное приложение”.

6. Нажать кнопку “Далее”.

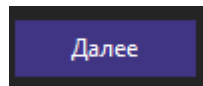


Рисунок 6. Кнопка “Далее”.

7. Появится окно настройки проекта.

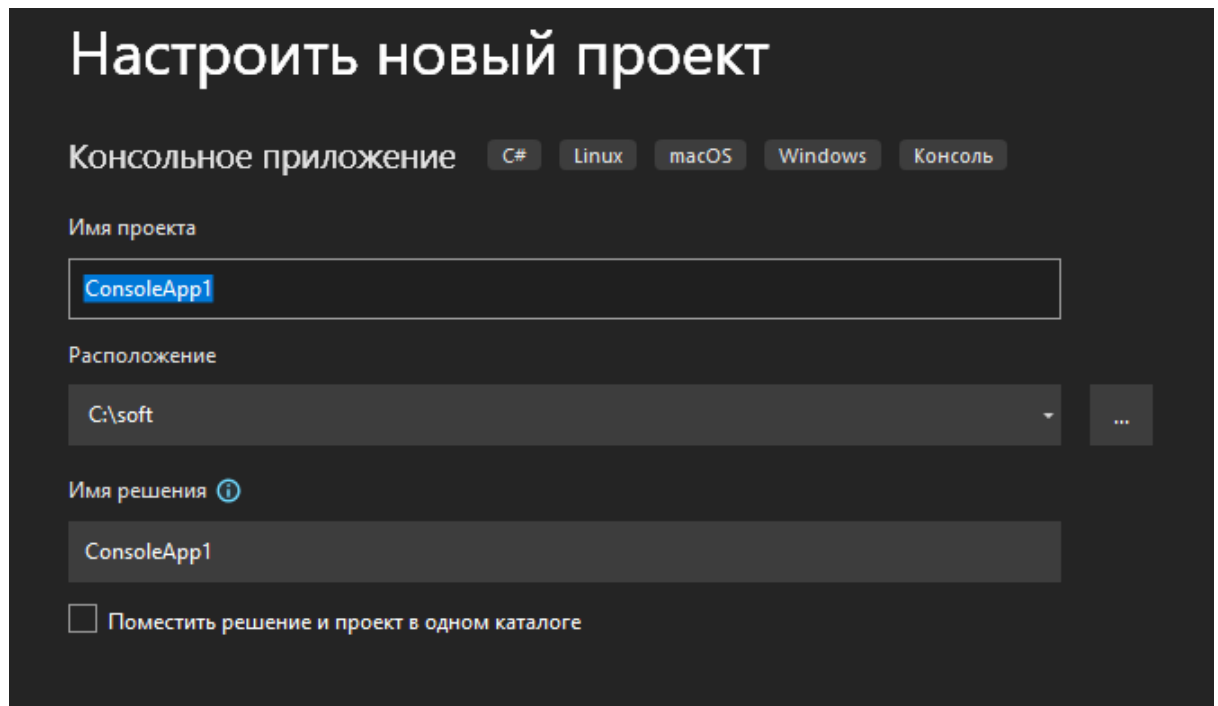


Рисунок 7. Окно настройки нового проекта.

8. Указать имя проекта. Желательно, чтобы оно было на латинице.

9. Указать желаемое расположение проекта.

10. Указать имя решения. Решение - это контейнер для одного или нескольких проектов в Visual Studio.

11. Нажать на кнопку “Далее”.

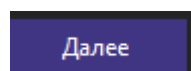


Рисунок 8. Кнопка “Далее”.

12. Откроется окно “Дополнительные сведения”. Выбрать версию .NET из предложенных. Желательно использовать последнюю доступную версию.

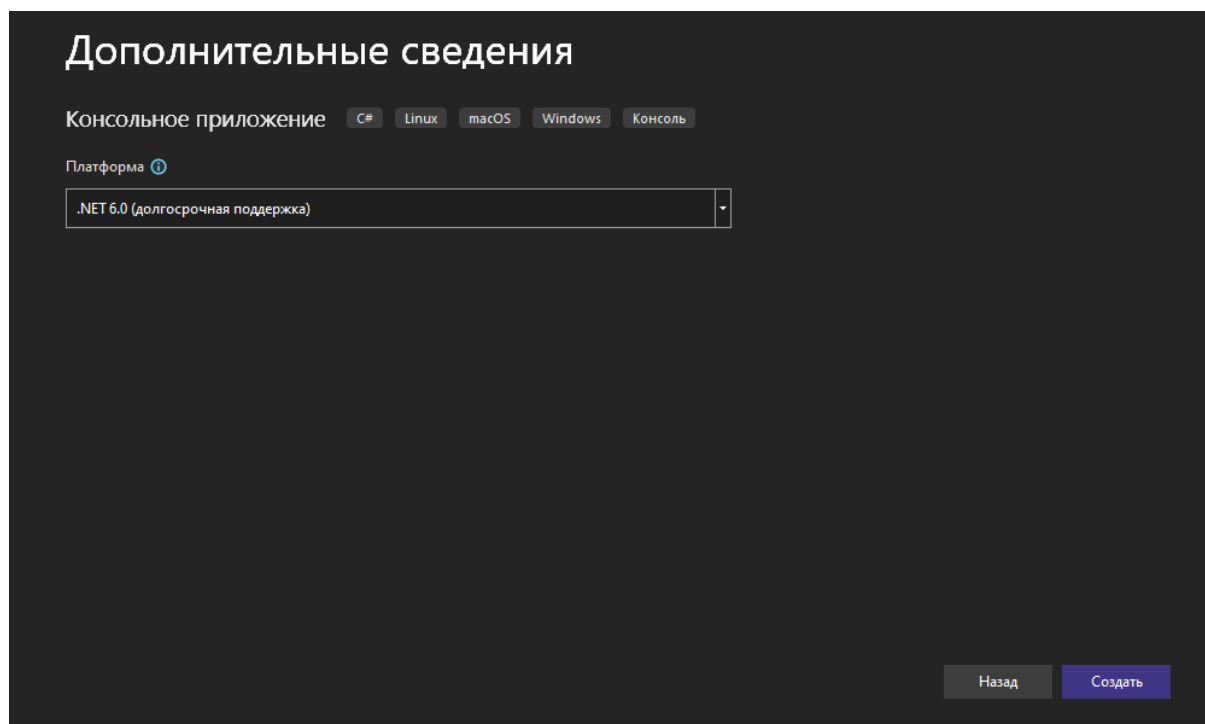


Рисунок 9. Окно дополнительных сведений нового проекта.

13. Нажать на кнопку “Создать”.

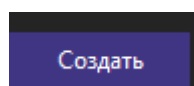


Рисунок 10. Кнопка “Создать”.

14. Откроется основное окно написания программы.

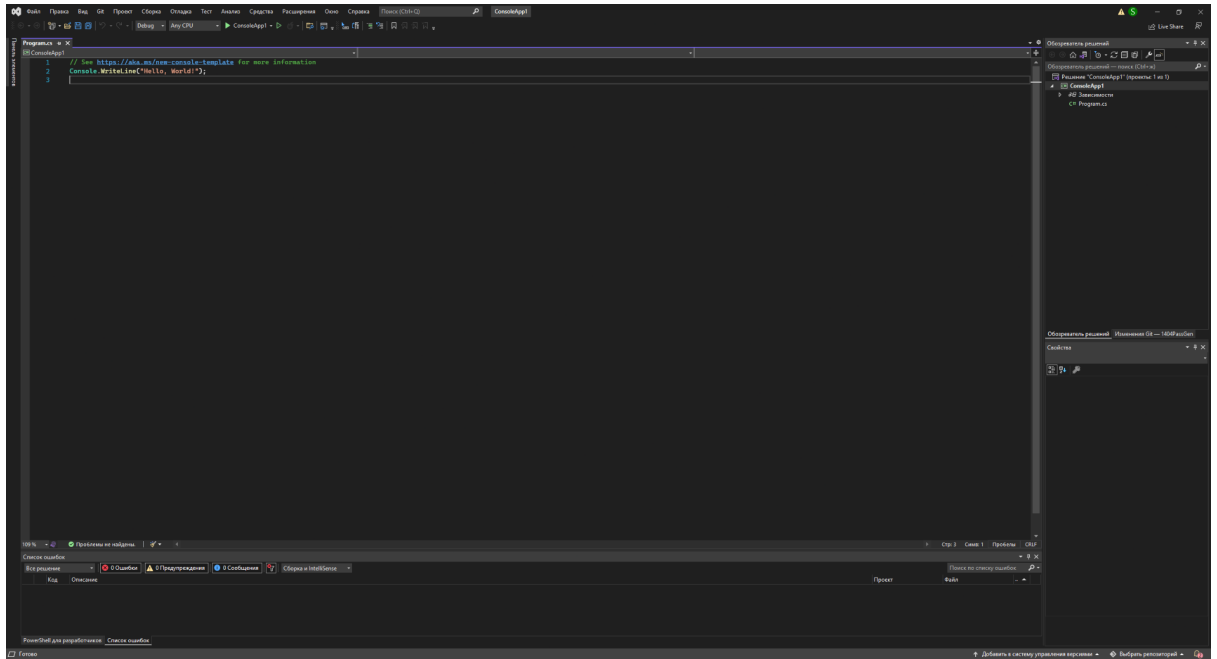


Рисунок 11. Основное окно Visual Studio 2022.

15. Нажать на кнопку “Проект” в верхнем правом углу.

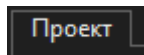


Рисунок 12. Вкладка настроек проекта.

16. В появившемся выпадающем списке нажать на кнопку “Управление пакетами NuGet...”.

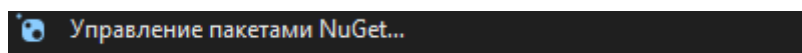


Рисунок 13. Кнопка управления проектами NuGet.

17. Откроется окно менеджера пакетов NuGet со списком всех установленных дополнительных пакетов.

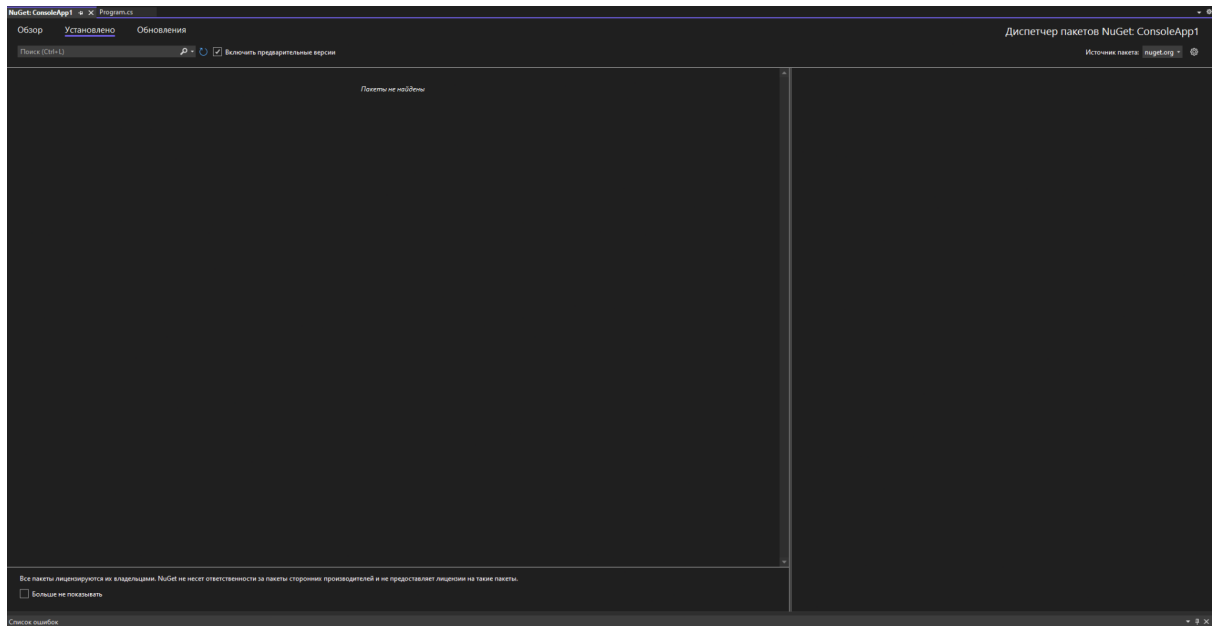


Рисунок 14. Окно менеджера пакетов Nuget.

18. Нажать на кнопку “Обзор”.

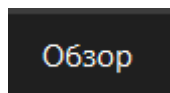


Рисунок 15. Вкладка “Обзор” в менеджере пакетов NuGet.

19. В поиске ввести “MySQL”.



Рисунок 16. Поиск в менеджере пакетов Nuget.

20. В результатах поиска будет пакет под названием “MySQL.Data”.

Нажать на него.



Рисунок 17. Библиотека MySQL.Data.

21. Справа откроется окно с информацией об этом пакете.

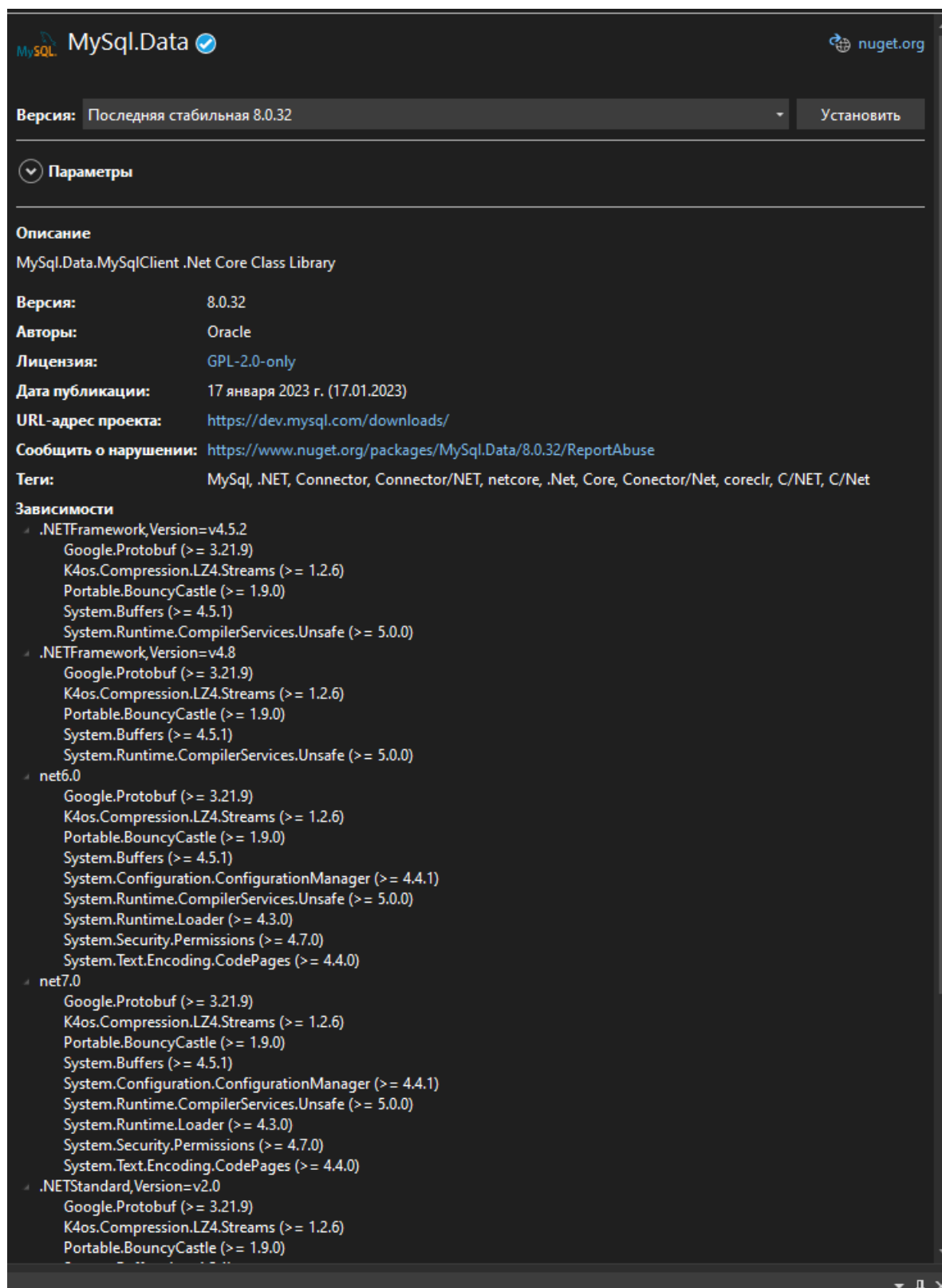


Рисунок 18. Описание пакета библиотеки MySQL.Data.

22. Нажать на кнопку “Установить”.



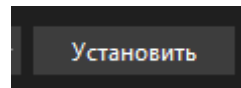


Рисунок 19. Кнопка “Установить”.

23. После начала установки появится окно справки вносимых изменений.

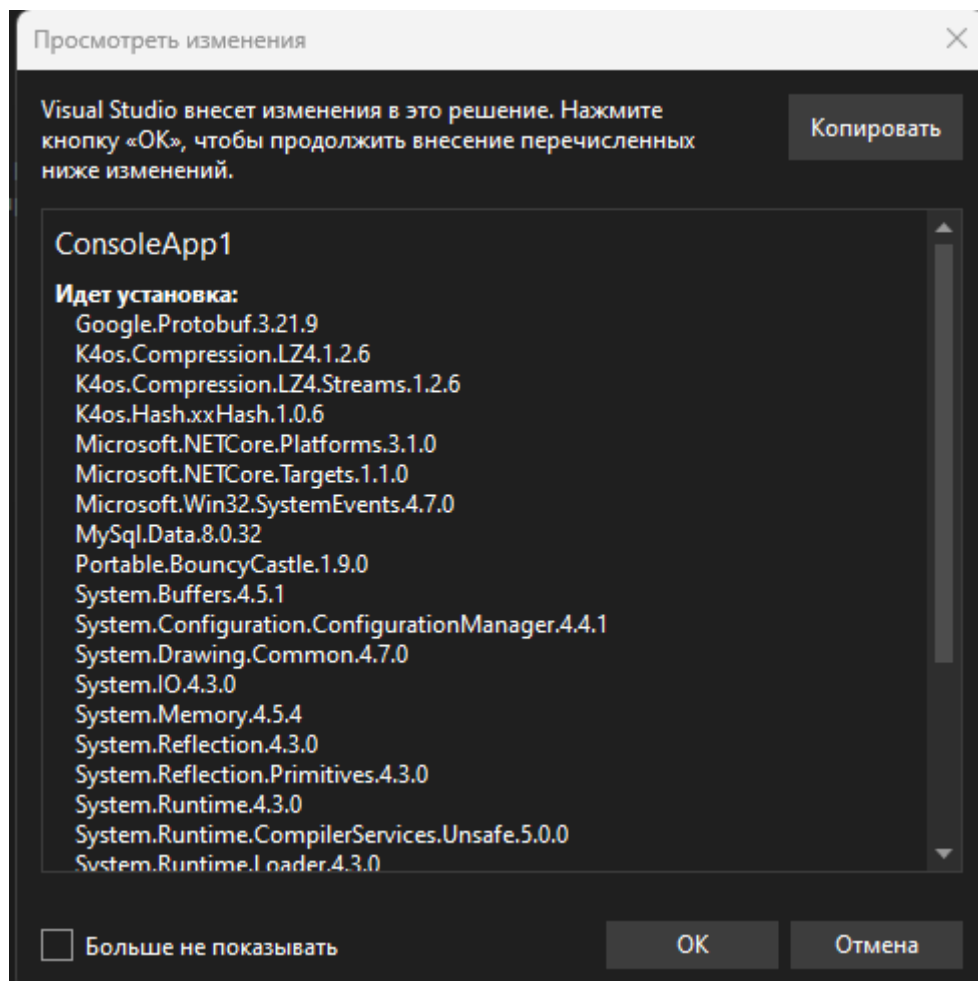


Рисунок 20. Список вносимых изменений.

24. Ознакомиться со справкой и нажать кнопку “Ок”.
25. Также появится окно “Принятие условий лицензионного соглашения”.

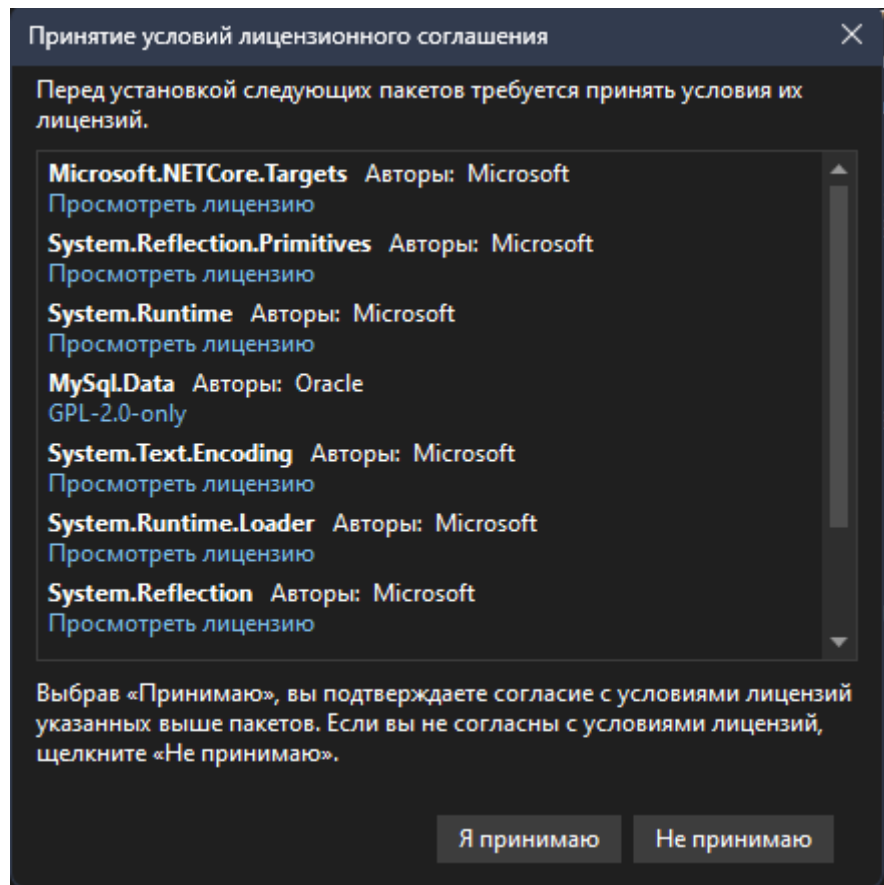


Рисунок 21. Список лицензионных соглашений.

26. Ознакомиться и нажать на кнопку “Я принимаю”.
27. Библиотека MySQL.Data установлена.
28. В верхнем левом углу окна менеджера пакетов NuGet нажать на крестик, чтобы закрыть его.

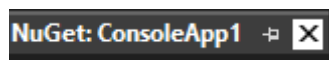


Рисунок 22. Закрытие вкладки менеджера пакетов NuGet.

29. Приступить к написанию кода.

```
using MySql.Data.MySqlClient;
```

Данная строка подключает библиотеку MySQL.Data к проекту.

“using” в C# — это конструкция языка, которая позволяет автоматически управлять объектами, реализующими интерфейс

IDisposable, такими как потоки ввода-вывода, базы данных, сокет и другие ресурсы, которые требуют очистки после использования.

Когда объект создается в блоке using, он автоматически привязывается к конструкции using, и когда блок using завершается, объект уничтожается автоматически, и вызывается метод Dispose(), который освобождает ресурсы, занятые этим объектом.

“IDisposable” — это интерфейс в C#, который позволяет управлять ресурсами, которые должны быть освобождены после использования объекта.

```
string connStr =  
"server=localhost;user=root;database=laba;port=3306;password=;charset  
=utf8;"
```

Это строка, в которую вводятся параметры подключения к базе данных.

server – это адрес сервера, на котором хранится база данных. В данном случае сервером выступает тот же компьютер, на котором база данных и хранится.

user – это имя учетной записи, которая будет использоваться для подключения. В данной работе будет использоваться учетная запись root.

database – это имя базы данных, по которой будет выполняться подключение.

port – это порт базы данных, по которому будет выполняться подключение.

password – пароль учетной записи пользователя, которая будет использоваться для подключения. В данном случае пароль отсутствует.

charset – это кодировка, которая будет использоваться.

```
MySqlConnection conn = new MySqlConnection(connStr);
```

MySqlConnection — это класс в библиотеке MySQL.Data, который представляет собой соединение с базой данных MySQL. Он используется для установления соединения с сервером MySQL и выполнения запросов к базе данных.

Выражение MySqlConnection conn = new MySqlConnection(connStr); создает новый объект класса MySqlConnection и присваивает его переменной conn. При создании объекта передается строка подключения connStr, которая определяет параметры соединения с базой данных

```
conn.Open();
```

conn.Open() — это метод объекта MySqlConnection, который используется для установления соединения с базой данных MySQL. После создания объекта MySqlConnection и настройки параметров соединения, необходимо вызвать метод Open(), чтобы установить соединение с базой данных. Метод Open() открывает соединение с базой данных, используя параметры, заданные в объекте MySqlConnection.

Подключение к базе данных выполнено!

```
conn.Close();
```

метод conn.Close() используется для закрытия соединения с базой данных MySQL. Когда соединение больше не нужно, его следует закрыть, чтобы освободить ресурсы сервера базы данных и позволить другим клиентам использовать его.

Кроме того, закрытие соединения гарантирует, что все незавершенные транзакции будут завершены, а все изменения, которые еще не были зафиксированы в базе данных, будут сохранены или отменены.

```
using MySql.Data.MySqlClient;
```

```
string connStr =
```

```
"server=localhost;user=root;database=laba;port=3306;password=;charset=utf8;"
```

```
MySqlConnection conn = new MySqlConnection(connStr);
```

```
conn.Open();
```

```
Console.WriteLine("Подключение к БД выполнено успешно!");
```

```
Console.ReadKey();
```

```
conn.Close();
```