|  |  |
| --- | --- |
| Департамент образования Ярославской области  Государственное профессиональное образовательное  автономное учреждение Ярославской области  «ЯРОСЛАВСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ им. Н.П. ПАСТУХОВА» | |
|  |  |
| **Отчет ПО УЧЕБНОЙ ПРАКТИКЕ**  **по профессиональному модулю ПМ.07**  **«Соадминистрирование и автоматизация БД и серверов»** | |
| ОУП ПМ.07.09.02.07.19ИП1.20 | |
|  | |
|  | |
|  | |
|  | |
|  | Студент  \_\_\_\_\_\_\_\_\_\_\_\_\_ Г.Г. Мымрин  «3» декабря 2022 г. |
|  | Преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_ Е.О. Куксов  «3» декабря 2022 г. |
|  |  |
| 2022 | |

**Содержание:**

[Введение 3](#_Toc117862231)

[Практическая работа № 1 5](#_Toc117862232)

[Практическая работа № 2 12](#_Toc117862233)

[Практическая работа № 3 17](#_Toc117862234)

[Практическая работа № 4 20](#_Toc117862235)

[Практическая работа № 5 24](#_Toc117862236)

[Практическая работа № 6 31](#_Toc117862237)

# **Введение**

Учебная практика направлена на:

­ формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта;

­ закрепление теоретических знаний, полученными студентами в процессе обучения профессиональных модулей;

­ углубление первоначального профессионального опыта студента, развития общих и профессиональных компетенций, проверку его готовности к самостоятельной трудовой деятельности;

­ сбор, систематизация и обобщение практического материала (в том числе) для использования в выпускной квалификационной работе.

По окончании практики студент сдаёт отчет в соответствии с содержанием тематического плана практики и по форме, установленной в ГПОАУ ЯО «Ярославский промышленно-экономический колледж имени Н.П. Пастухова».

Итоговая аттестация проводится в форме - дифференцированного зачёта.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями студент в ходе прохождения учебной практики должен:

Данные методические указания предназначены для студентов 4 курса специальности 09.02.07 «Информационные системы и программирование».

Методические указания содержат:

* требования к знаниям и умениям студентов;
* правила выполнения лабораторных работ (практических заданий);
* содержание лабораторных работ (практических заданий);
* контрольные вопросы;
* список использованной литературы.

В результате выполнения данного курса работ студент должен:

*Профессиональные компетенции*

ПК 7.1. Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов

ПК 7.2. Осуществлять администрирование отдельных компонент серверов.

ПК 7.3. Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.

ПК 7.4. Осуществлять администрирование баз данных в рамках своей компетенции.

ПК 7.5. Проводить аудит систем безопасности баз данных и серверов, с использованием регламентов по защите информации.

*Общие компетенции*

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения.

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

|  |  |  |
| --- | --- | --- |
| *Код*  *ПК, ОК* | *Умения* | *Знания* |
| ОК 1, 2, 3, 4, 5, 6, 7,8,9  ПК 7.1., 7.2., 7.3., 7.4., 7.5. | У1. проектировать и создавать базы данных;  выполнять запросы по обработке данных на языке SQL;  У2. осуществлять основные функции по администрированию баз данных;  У3. разрабатывать политику безопасности SQL сервера, базы данных и отдельных объектов базы данных;  У4. владеть технологиями проведения сертификации программного средства. | З1. модели данных, основные операции и ограничения;  З2. технологию установки и настройки сервера баз данных;  З3. требования к безопасности сервера базы данных;  З4. государственные стандарты и требования к обслуживанию баз данных |

# **Практическая работа № 1.1**

**Создание базы данных**

**Цель:** по заданной схеме данных разработать базу данных, создать и сохранить скрипт с БД.

**Задание:**

1. Изучите Пояснения к работе.
2. По диаграмме ERD, построенной на УП.05, с помощью СУБД MS SQL реализуйте базы данных.
3. Создайте скрипт базы данных.
4. Заполните базу данных первичными данными. (Если в базе данных нет таблицы с данными для авторизации, то добавьте ее).
5. Оформите отчет о проделанной работе (согласно плану отчета).

**Теоретический материал**

*База данных* - набор сведений, хранящихся некоторым упорядоченным способом.

*Система управления базами данных* — это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д.

*SQL* - язык структурированных запросов, основной задачей которого является предоставление простого способа считывания и записи информации в базу данных.

Простейшая схема работы с базой данных выглядит примерно так:

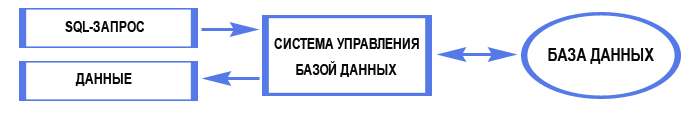


Рисунок 1. Схема работы с базой данных

В реляционной базе данных все данные можно представить в виде простых таблиц. Таблицы в реляционных базах данных обладают рядом свойств. Основными являются следующие:

* В таблице не может быть двух одинаковых строк.
* Столбцы располагаются в определенном порядке, который создается при создании таблицы. В таблице может не быть ни одной строки, но обязательно должен быть хотя бы один столбец.
* У каждого столбца есть уникальное имя (в пределах таблицы), и все значения в одном столбце имеют один тип (число, текст, дата...).
* На пересечении каждого столбца и строки может находиться только атомарное значение (одно значение, не состоящее из группы значений). Таблицы, удовлетворяющие этому условию, называют нормализованными.

*Первичный ключ* (сокращенно РК - primary key) - столбец, значения которого во всех строках различны.

В реляционных базах данных некоторая информация из одних таблиц присутствует в других, т.е. между ними имеются связи.

Всего существует 3 типа связей:

* [Один к одному](https://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh#onetoone);
* [Один ко многим](https://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh#onetomany);
* [Многие ко многим](https://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh#manytomany).

**Практическая часть**

По ERD (Рисунке 2) созданной раннее создадим базу данных в MS SQL (Рисунки 3 - )

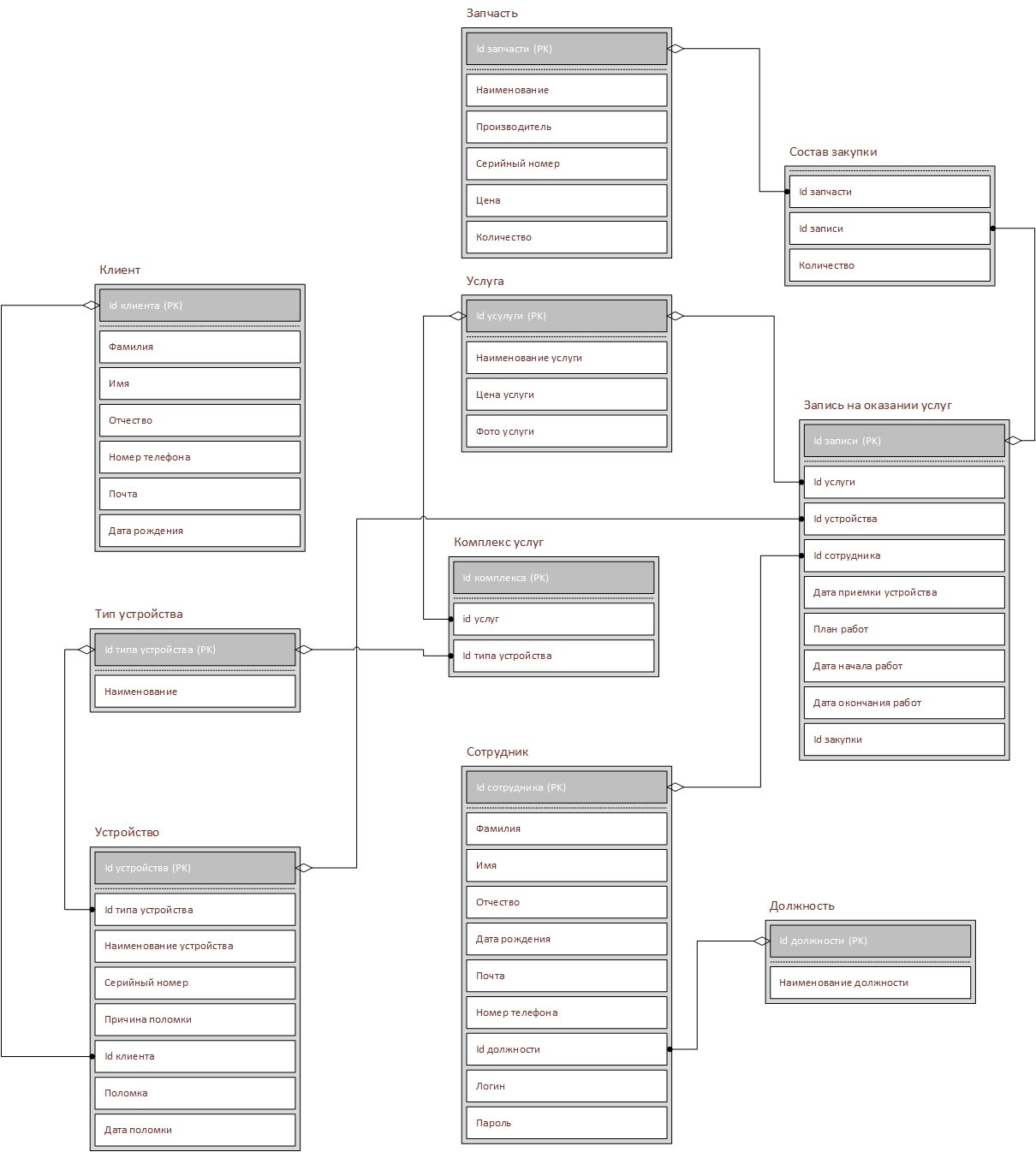


Рисунок 2 ERD

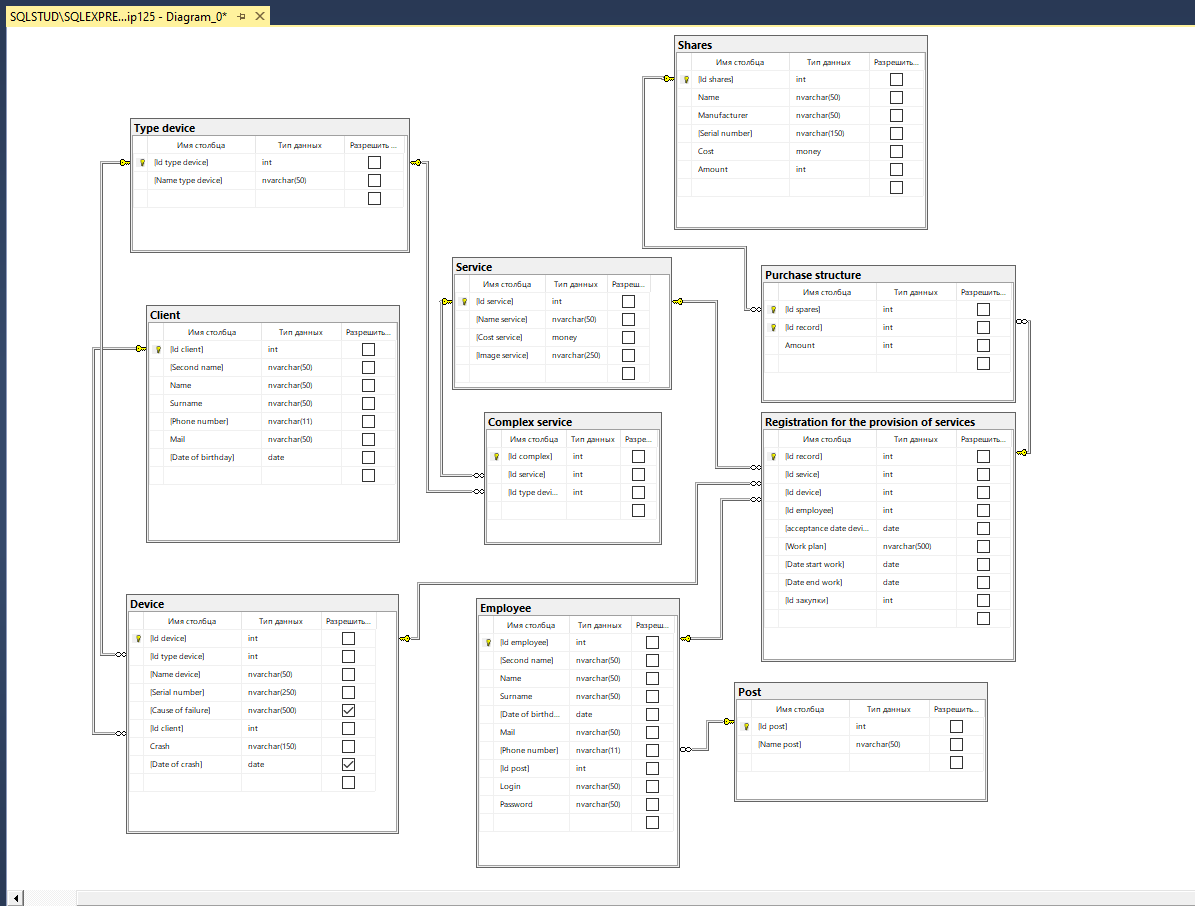


Рисунок 3 Диаграмма БД

Заполним БД тестовыми вариантами (Рисунок 4)

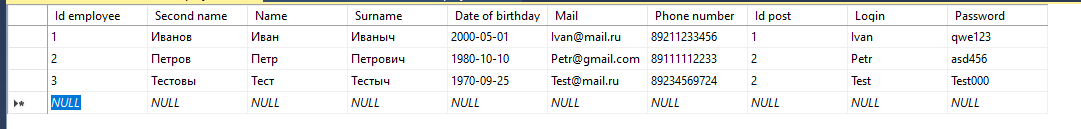


Рисунок 4 Тестовые данные

**Вывод:** в ходе проделанной работы, была создана ERD диаграмма и в MS SQL реализована сама БД.

**Контрольные вопросы**

1. Что такое «база данных»?

База данных - набор сведений, хранящихся некоторым упорядоченным способом.

1. Что такое первичный ключ (Primary key)?

Первичный ключ (сокращенно РК - primary key) - столбец, значения которого во всех строках различны.

1. Что такое внешний ключ (Foreign key)?

Внешние ключи позволяют установить связи между таблицами. Внешний ключ устанавливается для столбцов из зависимой, подчиненной таблицы, и указывает на один из столбцов из главной таблицы. Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы

1. Какие связи существуют в реляционных базах данных?

* Один к одному;
* Один ко многим;
* Многие ко многим.

1. Что подразумевается под целостностью данных?

Целостность базы данных — соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется ограничением целостности.

1. Что такое нормализация и каковы ее преимущества?

Нормализация – это процесс удаления избыточных данных. Благодаря ей можно избежать логических ошибок

# **Практическая работа № 1.2**

**Подключение базы данных.**

**Цель:** по заданной схеме данных разработать базу данных, создать и сохранить скрипт с БД, реализовать программное приложение с возможностью авторизации в системе.

**Задание:**

1. Изучите Пояснения к работе.
2. Создайте приложение (WPF) с помощью средств Visual Studio.
3. Оформите отчет о проделанной работе (согласно плану отчета).

**Теоретический материал**

*Windows Presentation Foundation* (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Преимущества WPF:

* Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения;
* Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET;
* Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением;
* Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.;
* Хорошее взаимодействие с WinForms, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms;
* Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое;
* Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной;
* Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10.

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше. Но это с лихвой компенсируется более широкими графическими возможностями и повышенной производительностью при отрисовке графики.

**Практическая часть**

Сперва добавим соединим бд с нашим приложение

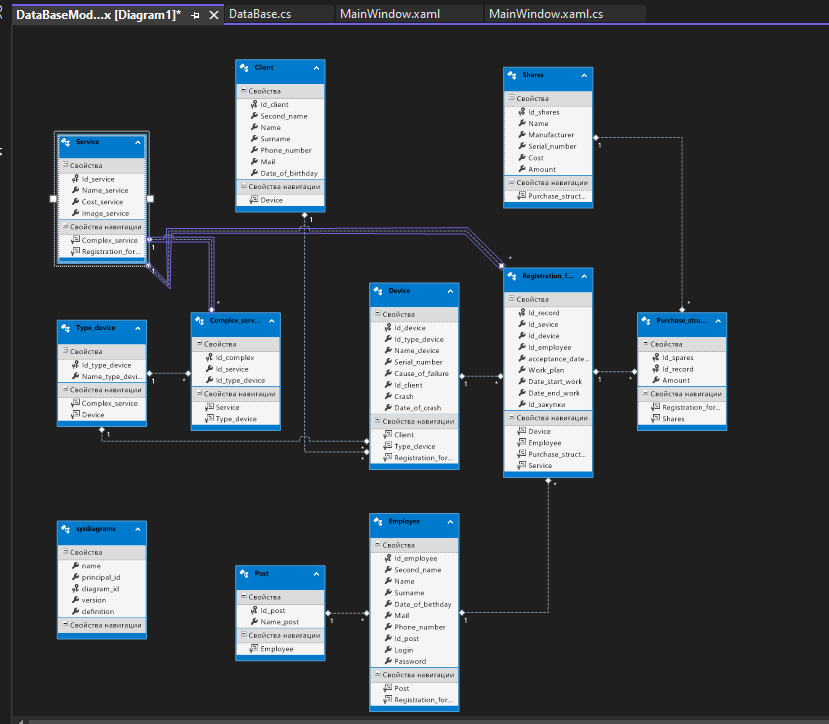


Рисунок 5. Схема БД

Далее создадим класс для подключения БД «Database» объявим переменную «\_context» объекта класса «Entities» и создадим публичный метод для возврата объекта и создания его, если его нет.

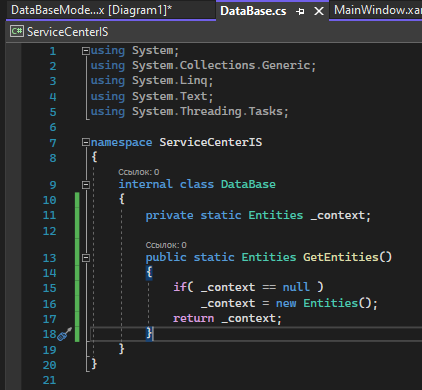


Рисунок 6 Класс DataBase

**Вывод:** в ходе проделанной работы, было выполнено соединение приложения с БД и создан класс для подключения БД.

**Контрольные вопросы**

1. Как создать подключение к базе данных?

Подключение к БД создается с помощь модели ADO

# **Практическая работа № 1.3**

**Создание руководства по стилю, логотипа, иконки. Размещение файлов в репозитории.**

**Цель:** разместить скрипт с базой данных, руководство по стилю, логотип и иконку в удаленный репозиторий проекта.

**Задание:**

1. Изучите Пояснения к работе.
2. Создайте руководство по стилю для своего будущего проекта.
3. Подберите или нарисуйте логотип и иконку для своей темы.
4. Разместите скрипт с базой данных, руководство по стилю и иконку в удаленный репозиторий на ветку «PR1\_3».
5. Оформите отчет о проделанной работе.

**Теоретическая часть**

Система контроля версий — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Git — одна из распределенных систем контроля версий.

GitHub — один из сервисов для использования системы контроля версий Git.

Git — одна из систем контроля версий.

Git предназначена, в основном, для работы распределенной команды разработчиков.

То есть разработчики могут находиться в разных концах света и работать над одним проектом.

Система Git очень экономична и не требует рассылки большого количества файлов. Отслеживаются и пересылаются изменения в файлах и ссылки на эти изменения. То есть основная рассылка — это рассылка разницы в ваших редактированиях.

Отсылаются только различия в папках и файлах. В любой момент времени вы можете возвратиться к тому или иному состоянию системы. Многие компании уделяют внимание хорошей и быстрой коммуникации между сотрудниками. В этом отношении, система контроля версий предоставляет большие возможности. Всю мощь и гибкость системы управления версиями вы сможете ощутить после изучения некоторого теоретического материала и применения на практике.

Основные обозначения:

* repository — некоторое хранилище файлов, ссылок на изменения в файлах
* commit — отслеживание изменений, сохраняет разницу в изменениях
* working directory — рабочий каталог на вашем компьютере
* staging area — область подготовленных файлов или рабочая область
* branch — ветка, состоит из набора коммитов, обычно ссылается на последний коммит
* merge — слияние, слияние веток в одну
* pull — втянуть, взять проект с сервера, получить изменения из удаленного репозитория
* push — вытолкнуть, отправить изменения на сервер

**Практическая часть:**

Git установлен уже на компьютере

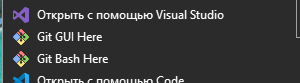
****

Рисунок 7 GIT

Создадим репозиторий

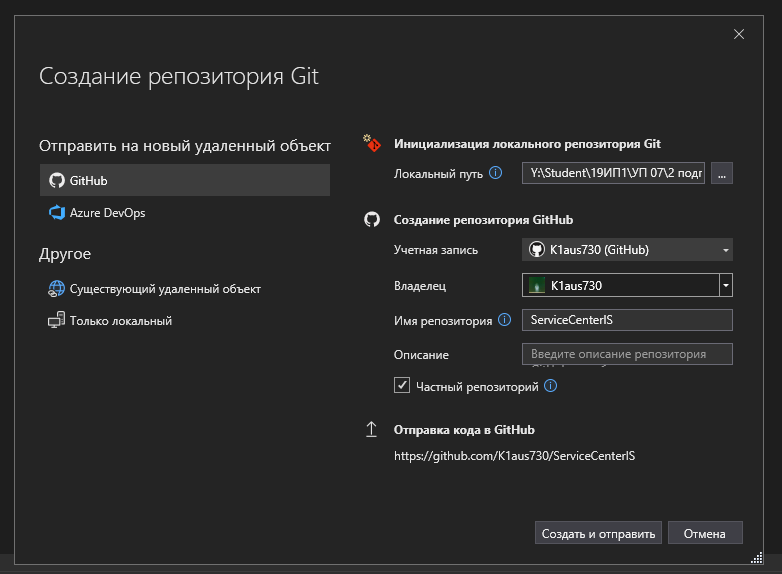


Рисунок 8 Создание репозитория

# **Практическая работа № 1.4**

**Создание словаря стилей. Создание Git репозитория с помощью Visual Studio.**

**Цель:** создать и настроить проект в Visual Studio, создать репозиторий, используя средства Visual Studio.

**Задание:**

1. Изучите Пояснения к работе.
2. Используя информацию в пункте «Пояснение к работе», а также руководство по стилю, создайте и настройте проект Visual Studio, реализуйте словарь ресурсов.
3. Создайте репозиторий согласно материалу, изложенному в «Пояснении к работе».

**Теоретическая часть**