**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет: Управления

Кафедра Информатики и информационных технологий

Направление подготовки / специальность: 09.03.03 Прикладная информатика

*(код, наименование)*

Направленность (профиль) / специализация: Корпоративные информационные системы управления

**Отчёт по лабораторным работам**

|  |  |
| --- | --- |
| по дисциплине: | « Программная инженерия» |
| на тему: |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **АВТОР** |
|  |  | Обучающийся(иеся) \_2\_ курса группы ИК-722 |
|  |  | \_\_\_заочной\_\_\_ формы обучения |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Хаванский Н.С.\_\_\_\_\_ |
|  |  | *(подпись) (фамилия, инициалы)* |
|  |  |  |
|  |  | **РУКОВОДИТЕЛЬ** |
|  |  |  |
|  |  | *(ученая степень, ученое звание)* |
|  |  | оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | « » 2025 г. |
|  |  | *(дата защиты)* |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Окулич.В.И\_\_\_\_ |
|  |  | *(подпись) (фамилия, инициалы)* |

Нижний Новгород, 2025г.

**Цель работы:**

Изучение и практическое освоение современных инструментов и технологий работы с системами контроля версий, экономическими документами и классификаторами, применяемыми в информационных системах. Формирование навыков анализа документов, построения баз данных и использования нотации BPMN для моделирования бизнес-процессов.

Инструменты:

Git — система контроля версий

GitHub / GitLab / GitFlick — онлайн-репозитории для хранения и управления кодом

SQL — язык запросов для работы с базами данных

СУБД (например, MariaBD) — для построения концептуальной, логической и физической моделей базы данных

Draw.io — для построения схем в нотации BPMN

DBeaver — универсальная графическая оболочка для работы с различными СУБД

Исходные данные:

Экономический документ (выбирается студентом, например: счет-фактура, накладная, акт, договор и т.п.)

Реквизиты документа — информация для анализа: финансовые, экономические, правовые поля

Два общероссийских классификатора (например: ОКВЭД, ОКАТО, ОКПО, ОКЕИ и т.д.)

Примеры команд Git и сравнение систем Git, GitHub, GitLab, GitFlick

Описание работы с документом и его жизненного цикла

1. Освоение работы в Git, GitHab, GitLab.
   1. Назначение, возможности, особенности.
   2. Сравнение (Git, GitHub vs GitLab).
   3. Российская система GitFlick.
   4. Команды системы Git: названия и назначение.
   5. Зарегистрироваться в одном из репозиториев и записать в них отчёты по лабораторным работам.

Освоение работы в Git, GitHub, GitLab

1. Назначение, возможности, особенности

Git – распределённая система контроля версий (VCS), которая позволяет:

* Отслеживать изменения в коде.
* Совместно работать над проектами.
* Возвращаться к предыдущим версиям.
* Создавать ветки (branches) для параллельной разработки.

GitHub – облачная платформа для хостинга Git-репозиториев с дополнительными функциями:

* Pull Requests (запросы на слияние кода).
* Issues (отслеживание задач и багов).
* Actions (CI/CD).
* Wiki, Projects (управление проектами).

GitLab – альтернатива GitHub с похожим функционалом, но с возможностью:

* Самостоятельного развёртывания (self-hosted).
* Встроенного CI/CD (без сторонних инструментов).
* Более гибкой настройки для корпоративных решений.

1. Сравнение Git, GitHub GitLab

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Git (локальный)** | **GitHub (облачный)** | **GitLab (облачный/локальный)** |
| **Тип** | Система контроля версий | Хостинг Git-репозиториев | Альтернативный хостинг с CI/CD |
| **Доступность** | Устанавливается локально | Только облако | Облако + можно развернуть у себя |
| **Бесплатные приватные репозитории** | – | Да (с ограничениями) | Да (полноценные) |
| **CI/CD** | Требует настройки | GitHub Actions (удобно, но не так гибко) | Встроенный мощный CI/CD |
| **Интеграции** | Любые через хуки | Много сторонних сервисов(API) | Гибкие, особенно для DevOps |
| **Российские аналоги** | – | – | GitFlick (полный аналог, но российский) |

1. Российская система GitFlick

GitFlick – российский аналог GitHub/GitLab, разработанный для:

* Хостинга кода внутри РФ.
* Соответствия законам о данных.
* Альтернативы западным сервисам.

Почти полное импортозамещение GitLab.

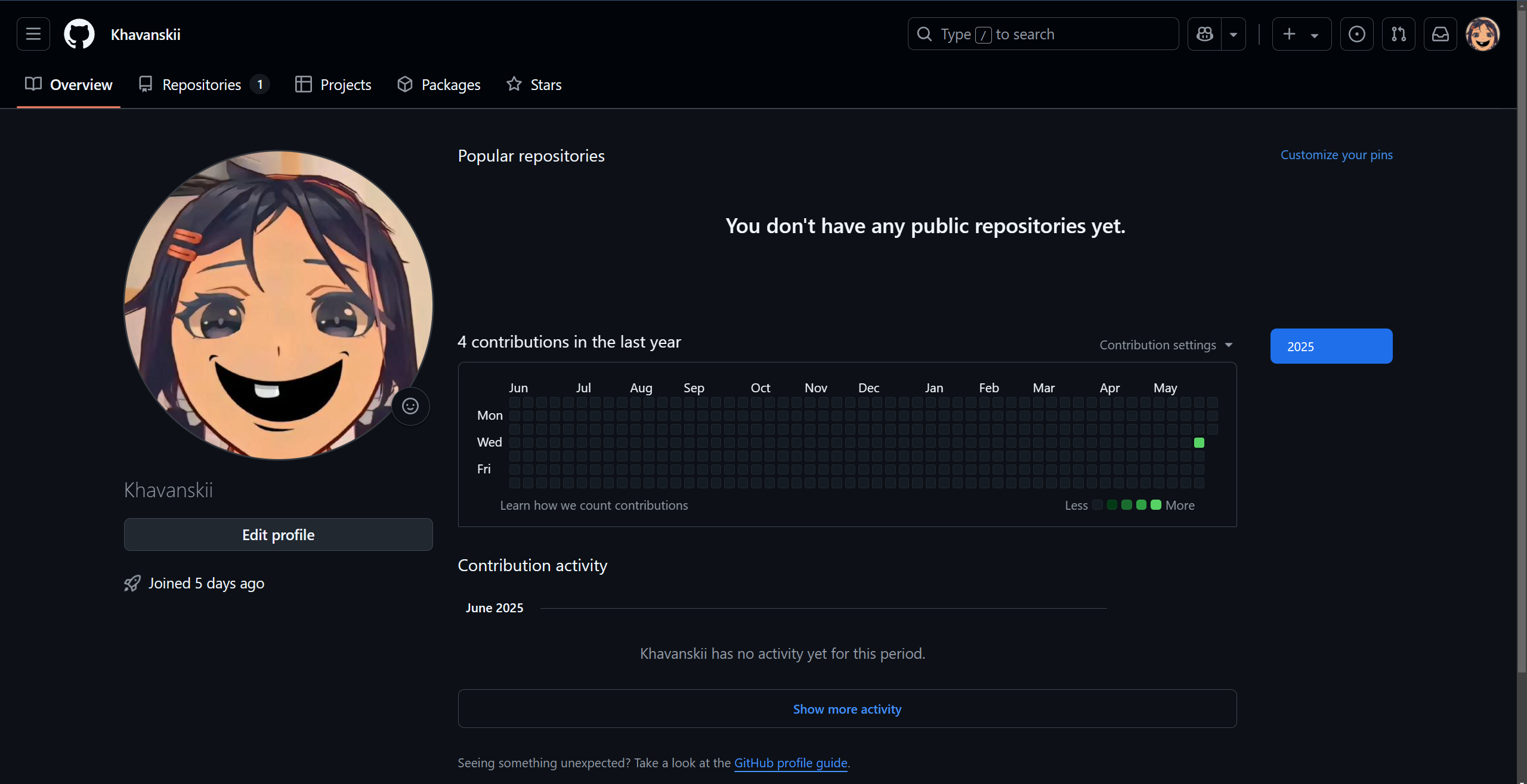
1. Основные команды Git

|  |  |
| --- | --- |
| Команда | Назначение |
| git init | Создать новый репозиторий |
| git clone <url> | Склонировать удалённый репозиторий |
| git add <file> | Добавить файлы в индекс (staging) |
| git commit -m "текст" | Закоммитить изменения |
| git push | Отправить изменения на сервер |
| git pull | Загрузить изменения с сервера |
| git branch | Показать ветки |
| git checkout <branch> | Переключиться на ветку |
| git merge <branch> | Слить ветки |
| git status | Показать состояние репозитория |

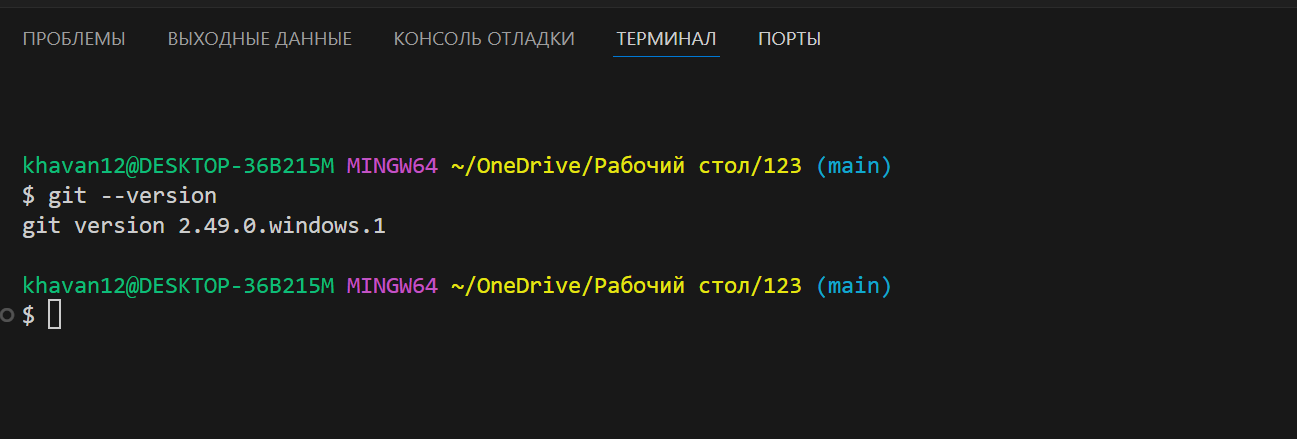
Зарегистрироваться в одном из репозиториев и записать в них

отчёты по лабораторным работам.

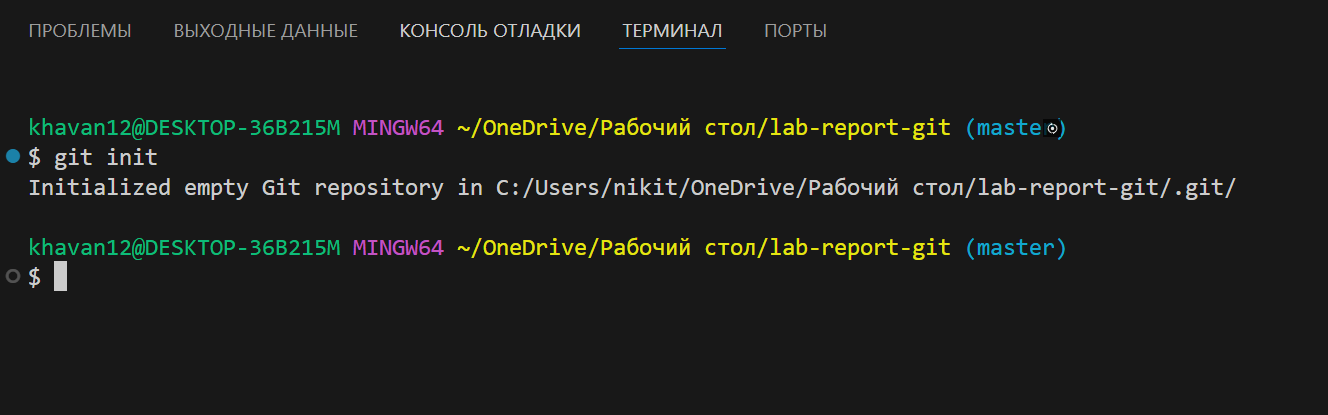
1. Регистрация на GitHub



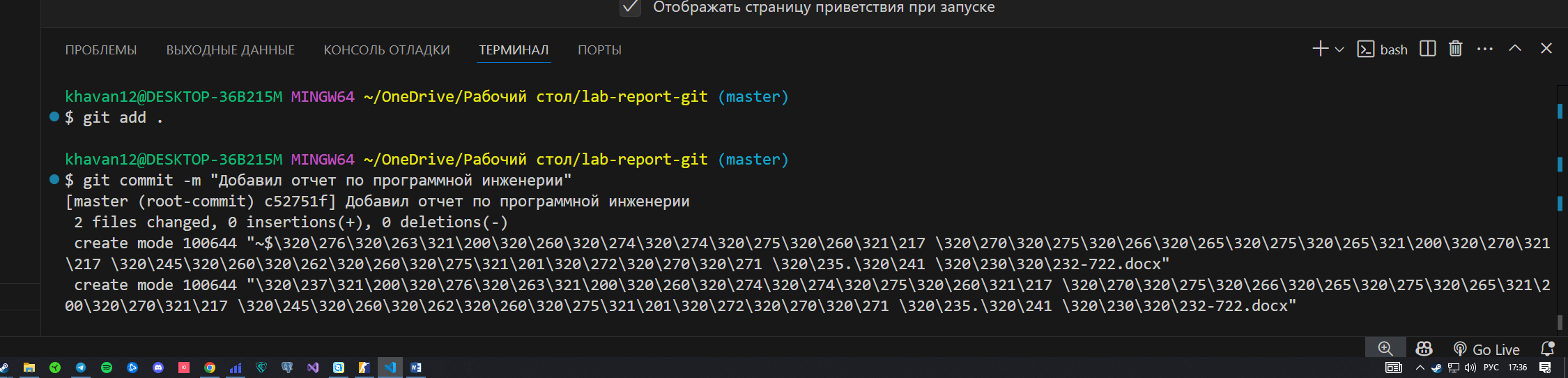
Профиль на GitHub



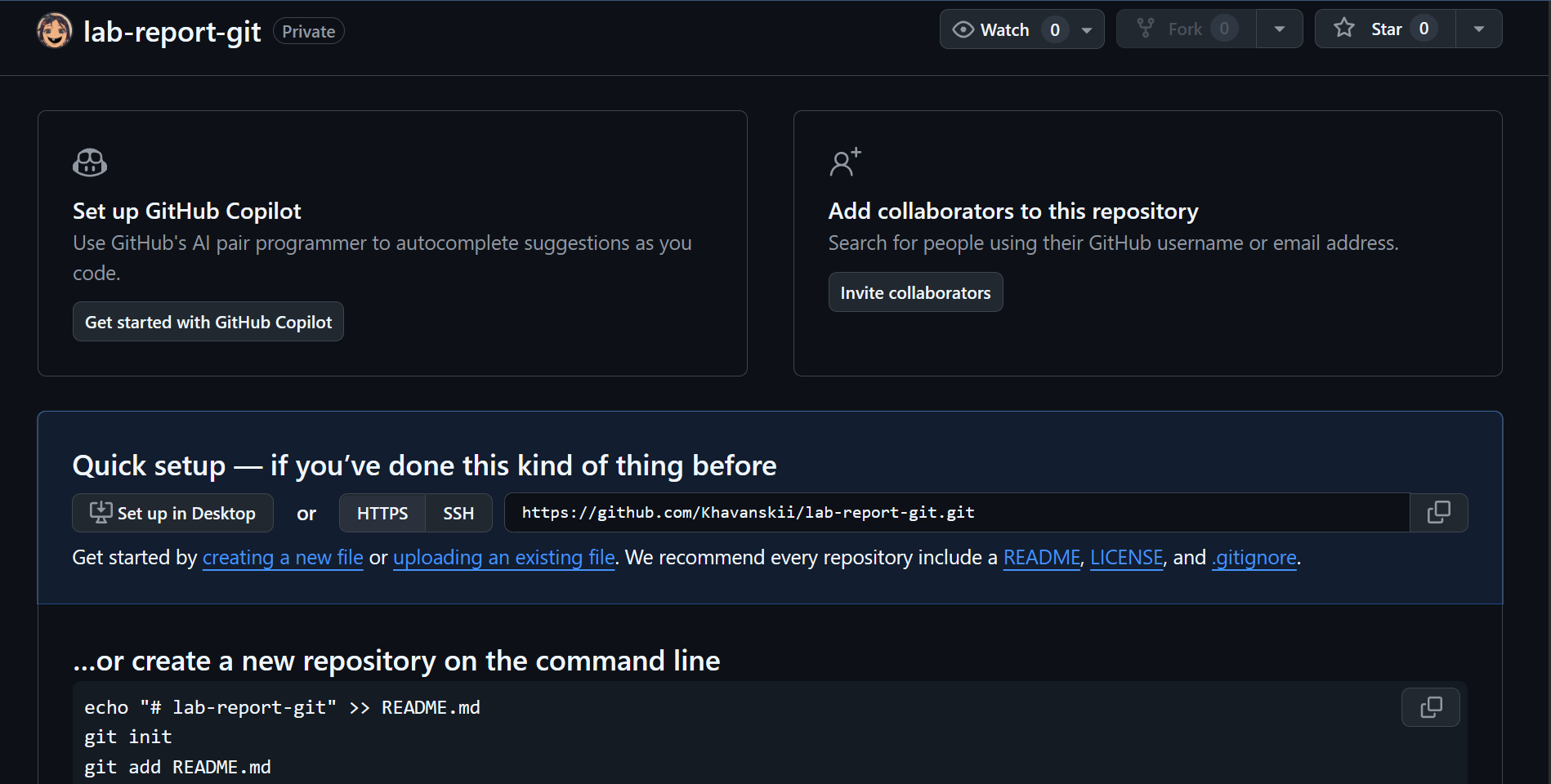
Git команды в VS Code



Создание git - репотозитория



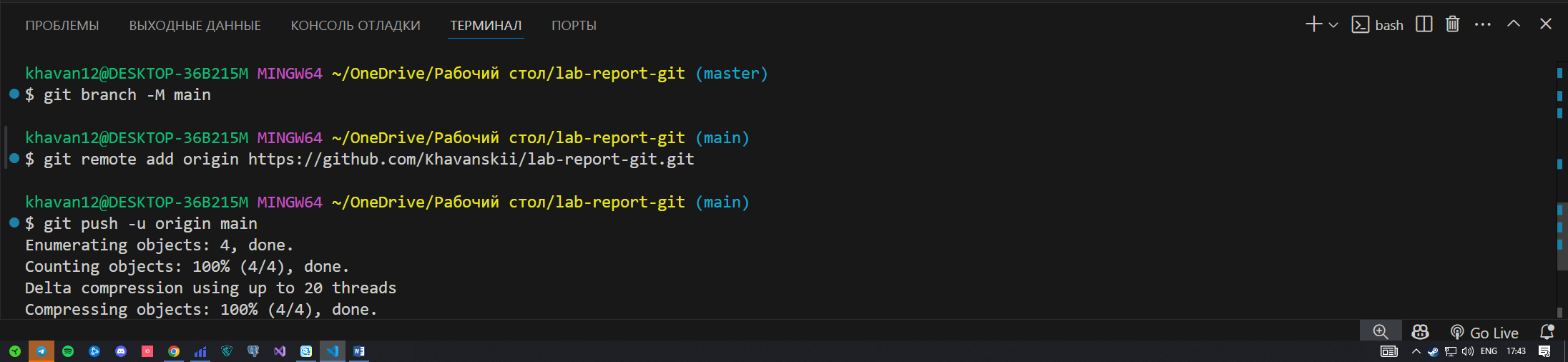
Добавление его в гит



Создание репозитория на GitHub

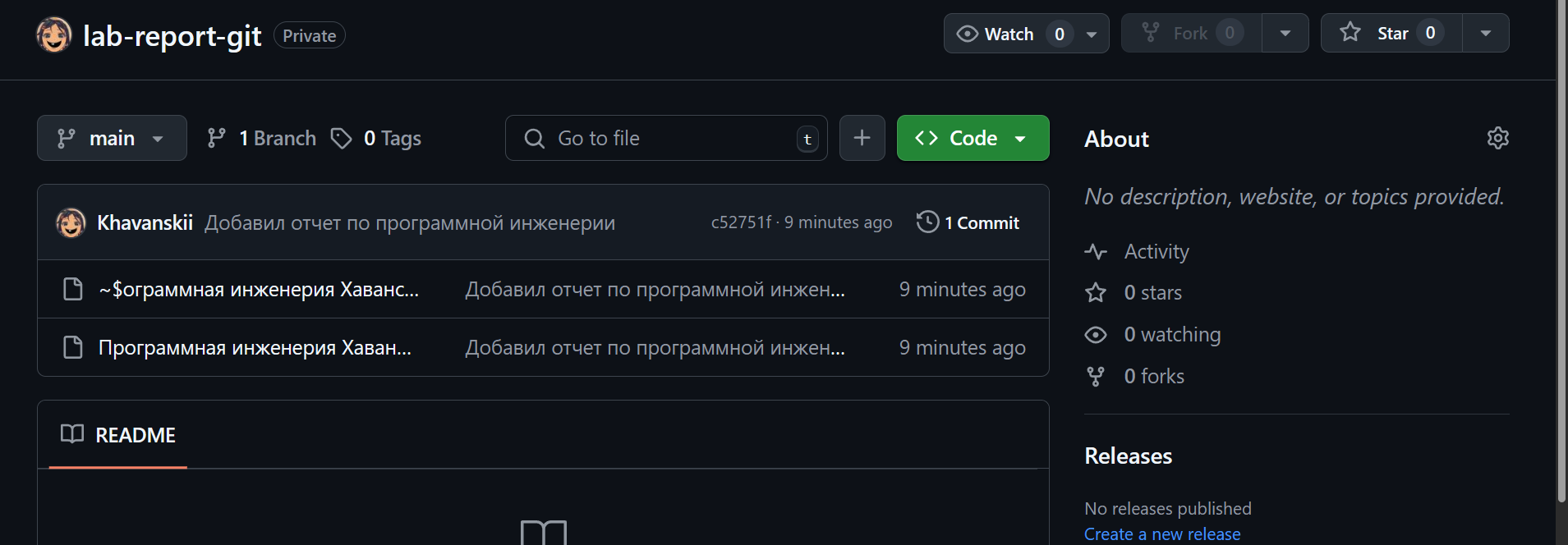


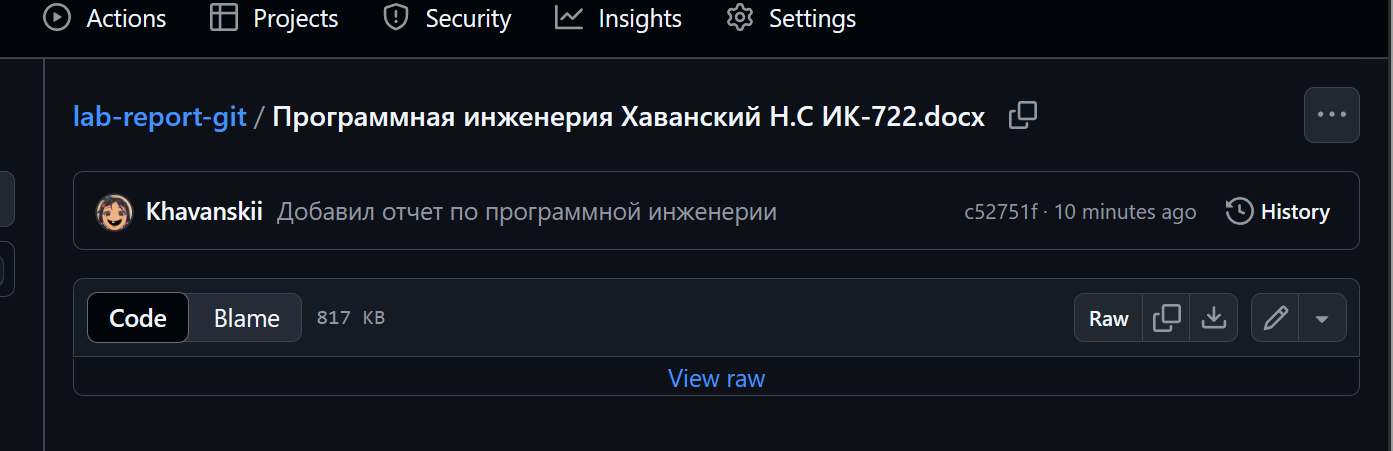
Создание репозитория на GitHub



Связывание VS Code и GitHub

Итоги работы:  

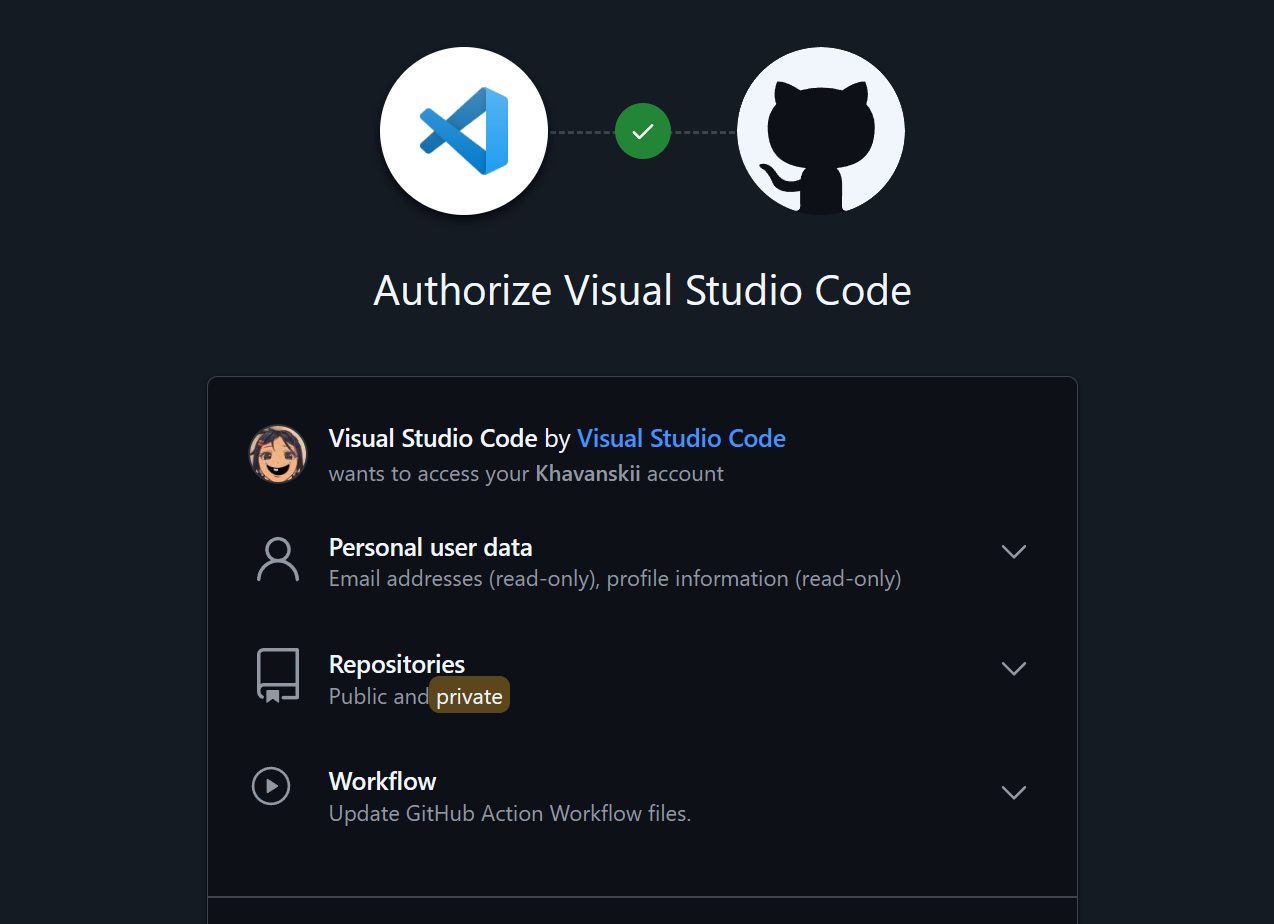


Итог:  
В ходе выполнения лабораторной работы были выполнены следующие шаги:

Создана рабочая папка lab-report-git и помещён в неё файл отчёта:  
**«Программная инженерия Хаванский Н.С ИК-722.docx»**

Папка была открыта в **VS Code**, где инициализирован локальный Git-репозиторий командой git init.

Создан репозиторий на **GitHub**, выполнена привязка (git remote add origin) и отправка файлов на удалённый сервер (git push).



Анализ кассового чека и создание БД

На примере чека ИП Якушенко С.В. (компьютерный клуб "Джаст Проджект")

b. Реквизитный анализ чека

**Общие реквизиты:**

Номер чека: 40

Дата/время: 09.05.25 14:15

Кассир: Якушенко Сергей Валерьевич

Смена: 309

**Финансовые реквизиты:**

Сумма: 50.00 ₽

Форма оплаты: БЕЗНАЛИЧНЫМИ

НДС: БЕЗ НДС

Тип операции: ПРИХОД

**Экономические реквизиты:**

Услуга: Аренда ПК

Количество: 1 шт

Цена: 50.00 ₽

**Технические реквизиты:**

ККТ: 0007584905029170

ФПД: 2584487982

ФН: 7380440700212846

ФД: 33956

ИНН: 526333007377

Адрес: г. Нижний Новгород, пр-кт Тагарина, д.29

**c. Классификация реквизитов**

|  |  |  |  |
| --- | --- | --- | --- |
| Реквизит | Основание | Признак 1 (тип) | Признак 2 (обязательность) |
| Сумма | Стоимость услуги | DECIMAL(10,2) | Обязательный |
| Форма оплаты | Способ расчета | VARCHAR(20) | Обязательный |
| НДС | Налоговая ставка | VARCHAR(10) | Обязательный |
| Услуга | Наименование предмета расчета | VARCHAR(100) | Обязательный |
| Количество | Объем услуги | INT | Обязательный |
| Цена | Стоимость единицы | DECIMAL(10,2) | Обязательный |

**d. Создание БД для учета чеков**

-- Торговые точки

CREATE TABLE Outlet (

OutletID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL, -- "Джаст Проджект"

Address VARCHAR(200) NOT NULL,

INN VARCHAR(12) NOT NULL

);

-- Кассы

CREATE TABLE KKT (

KKTID INT PRIMARY KEY AUTO\_INCREMENT,

OutletID INT NOT NULL,

RegNumber VARCHAR(20) NOT NULL, -- "0007584905029170"

FOREIGN KEY (OutletID) REFERENCES Outlet(OutletID)

);

-- Чеки

CREATE TABLE Receipt (

ReceiptID INT PRIMARY KEY AUTO\_INCREMENT,

KKTID INT NOT NULL,

Number VARCHAR(50) NOT NULL, -- "40"

DateTime DATETIME NOT NULL, -- "2025-05-09 14:15:00"

ShiftNumber INT NOT NULL, -- "309"

Cashier VARCHAR(100) NOT NULL, -- "Якушенко С.В."

Total DECIMAL(10,2) NOT NULL, -- "50.00"

PaymentMethod VARCHAR(20), -- "БЕЗНАЛИЧНЫМИ"

FPD VARCHAR(20) NOT NULL, -- "2584487982"

FD VARCHAR(20) NOT NULL, -- "33956"

FOREIGN KEY (KKTID) REFERENCES KKT(KKTID)

);

-- Позиции чеков

CREATE TABLE ReceiptItem (

ItemID INT PRIMARY KEY AUTO\_INCREMENT,

ReceiptID INT NOT NULL,

ProductName VARCHAR(100) NOT NULL, -- "Аренда ПК"

Quantity INT NOT NULL, -- "1"

Price DECIMAL(10,2) NOT NULL, -- "50.00"

VAT VARCHAR(10) NOT NULL, -- "БЕЗ НДС"

ItemType VARCHAR(20) NOT NULL, -- "УСЛУГА"

FOREIGN KEY (ReceiptID) REFERENCES Receipt(ReceiptID)

);

**Пример заполнения БД**

-- Торговая точка

INSERT INTO Outlet (Name, Address, INN)

VALUES ('компьютерный клуб Джаст Проджект', 'г. Нижний Новгород, пр-кт Тагарина, д.29', '526333007377');

-- Касса

INSERT INTO KKT (OutletID, RegNumber)

VALUES (1, '0007584905029170');

-- Чек

INSERT INTO Receipt (KKTID, Number, DateTime, ShiftNumber, Cashier, Total, PaymentMethod, FPD, FD)

VALUES (1, '40', '2025-05-09 14:15:00', 309, 'Якушенко Сергей Валерьевич', 50.00, 'БЕЗНАЛИЧНЫМИ', '2584487982', '33956');

-- Позиция чека

INSERT INTO ReceiptItem (ReceiptID, ProductName, Quantity, Price, VAT, ItemType)

VALUES (1, 'Аренда ПК', 1, 50.00, 'БЕЗ НДС', 'УСЛУГА');

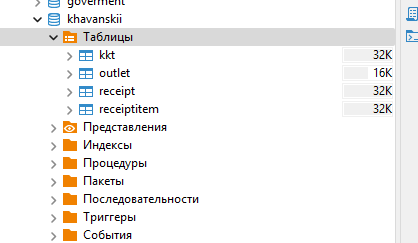
**Проверка данных:**

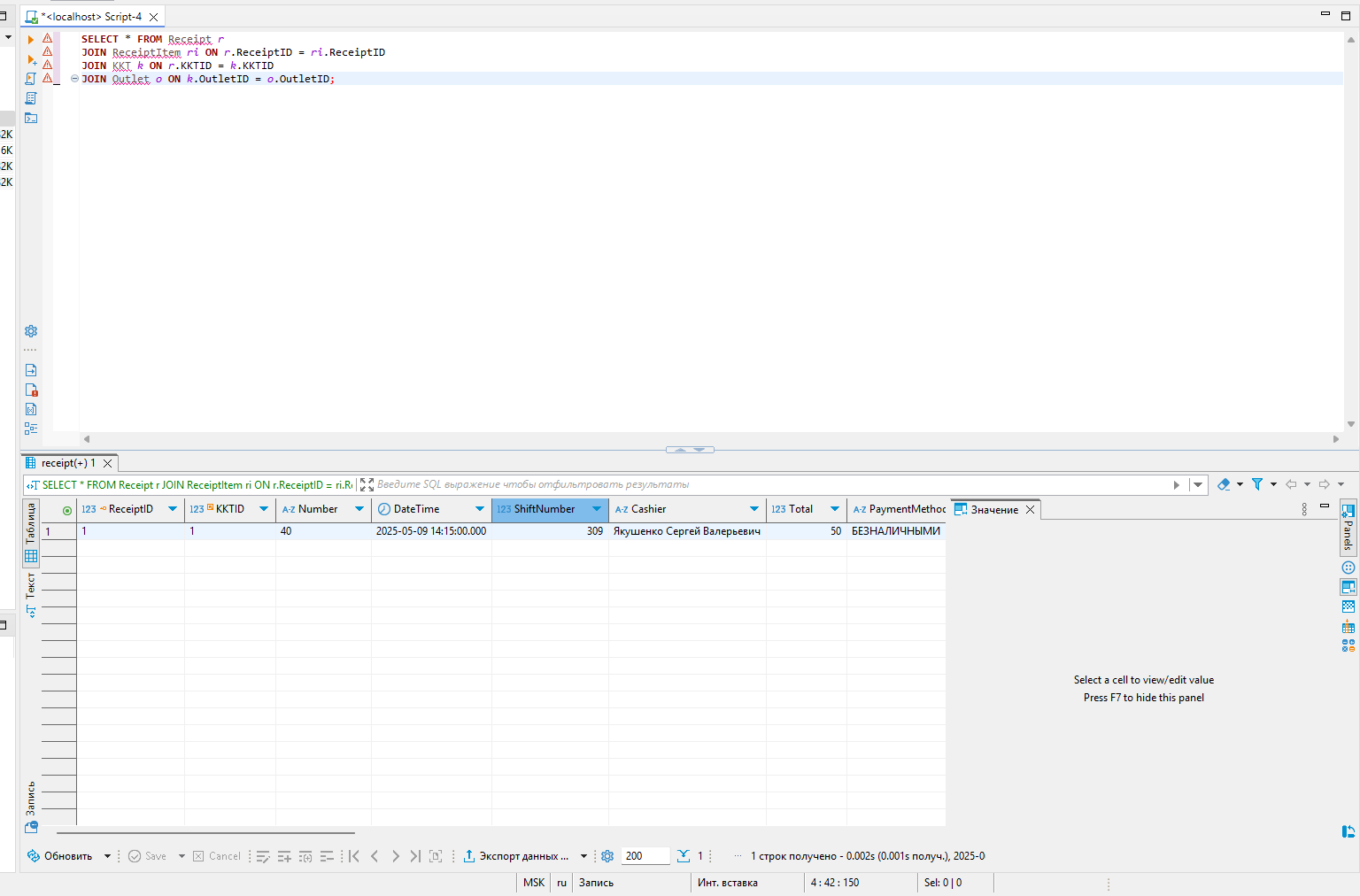
SELECT \* FROM Receipt r

JOIN ReceiptItem ri ON r.ReceiptID = ri.ReceiptID

JOIN KKT k ON r.KKTID = k.KKTID

JOIN Outlet o ON k.OutletID = o.OutletID;

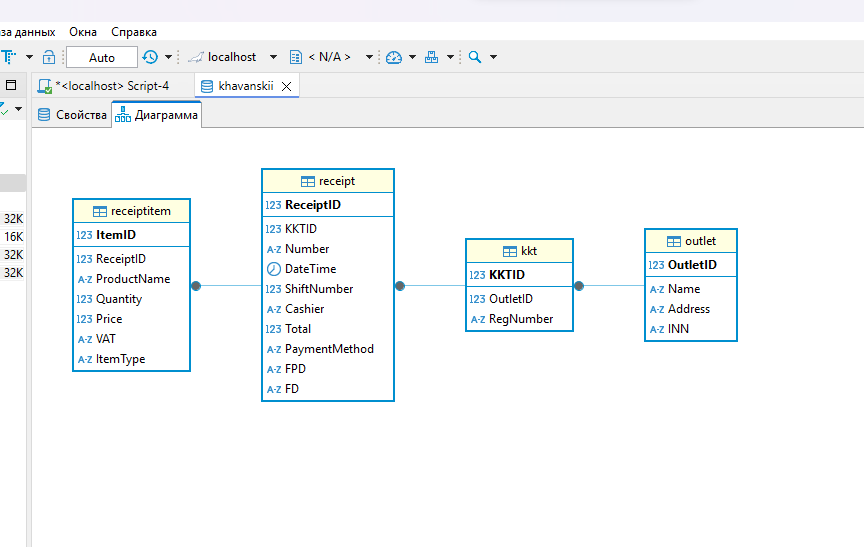
****

****

**e. Процесс работы с чеком (BPMN)**

Описание процесса:

1. Формирование чека на кассе
2. Фискализация (подписание ФН)
3. Отправка в ОФД
4. Сохранение в БД
5. Выдача чека покупателю

****

Пояснение этапов:

Формирование чека:

Кассир добавляет позиции (услуги/товары)

Система рассчитывает итоговую сумму

Фискализация:

Данные передаются в фискальный накопитель (ФН)

Генерируется фискальный признак (ФПД)

Отправка в ОФД:

Чек автоматически отправляется оператору фискальных данных

ОФД возвращает подтверждение

Сохранение в БД:

Все данные чека записываются в базу данных

Связываются с торговой точкой и кассой

Выдача покупателю:

Печать бумажного чека

Отправка электронной копии на email

3.Исследование общероссийских классификаторов (по книге Грекула).

a. Типы классификаторов, назначение

b. О кодировании классификаторов (формула классификатора)

c. Характеристики классификаторов для двух выбранных классификаторов.

d. Схема БД для выбранных классификаторов.

1. Типы классификаторов и их назначение

Классификатор — это систематизированный перечень объектов (товаров, услуг, организаций и т.д.), каждому из которых присвоен уникальный код.

Основные типы классификаторов в РФ:

|  |  |  |
| --- | --- | --- |
| Тип классификатора | Назначение | Пример |
| ОКПД2 (ОК 034-2014) | Классификация продукции и услуг для госзакупок, статистики, налогообложения | "27.90.10 — Услуги по аренде ПК" |
| ОКВЭД2 (ОК 029-2014) | Кодирование видов экономической деятельности | "63.11 — Деятельность по обработке данных" |
| ОКАТО (ОК 019-95) | Административно-территориальное деление | "22 440 — Нижний Новгород" |
| ОКУД (ОК 011-93) | Унификация форм документов (бланков, отчётов) | "0301005 — Кассовый чек" |
| ОКФС (ОК 027-99) | Кодирование форм собственности | "16 — Частная собственность" |

Назначение:

Упрощение автоматизированного учёта.

Стандартизация данных для госорганов (ФНС, Росстат).

Совместимость между информационными системами.

1. Формула классификатора (кодирование)

Классификаторы строятся по иерархической системе кодирования:

Код = X + Y + Z + ...

X — раздел (1-2 знака).

Y — группа (2-3 знака).

Z — подгруппа/вид (дополнительные знаки).

Пример для ОКПД2:

27.90.10 — Услуги по аренде компьютеров:

27 — Раздел: IT-услуги.

90 — Группа: Техническое обслуживание.

10 — Подгруппа: Аренда ПК.

1. Характеристики двух классификаторов

1. ОКПД2 (ОК 034-2014)

Назначение: Классификация товаров и услуг для госзакупок (44-ФЗ).

Структура кода: 6-9 цифр (XX.XX.XX).

Пример:

Код: 27.90.10

Наименование: "Услуги по аренде компьютеров".

2. ОКВЭД2 (ОК 029-2014)

Назначение: Определение вида деятельности компании (указывается в ЕГРЮЛ).

Структура кода: 2-6 цифр (XX.XX).

Пример:

Код: 63.11

Наименование: "Деятельность по обработке данных".

1. Схема БД для классификаторов

-- Таблица классификаторов

CREATE TABLE Classifier (

ClassifierID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50) NOT NULL, -- "ОКПД2", "ОКВЭД2"

Description TEXT

);

-- Таблица кодов

CREATE TABLE ClassifierCode (

CodeID INT PRIMARY KEY AUTO\_INCREMENT,

ClassifierID INT NOT NULL,

Code VARCHAR(20) NOT NULL, -- "27.90.10"

Name VARCHAR(200) NOT NULL, -- "Услуги по аренде ПК"

Level INT NOT NULL, -- Уровень иерархии (1-3)

FOREIGN KEY (ClassifierID) REFERENCES Classifier(ClassifierID)

);

-- Таблица объектов (например, товаров/услуг)

CREATE TABLE Product (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

CodeID INT, -- Ссылка на код классификатора

FOREIGN KEY (CodeID) REFERENCES ClassifierCode(CodeID)

);

