

## Отчет по лабораторной работе № 3

Тема: подключение модели Ollama к редакторам кода

Тестируемая модель: qwen2.5-coder:7b.


Тестируемые редакторы и плагины: JetBrains IDEA, DevovxxGenie

### 1. Установка плагина и настройка конфига

DevovxxGenie можно установить как внутри самого JetBrains IDEA так и скачать с сайта и установить вручную указав путь до архива:

Code Tools

Machine Learning



DevovxxGenie

★★★★★

Devovxx

Overview

Versions

Reviews

Compatible with IntelliJ

Plugin Versions

Compatibility: IntelliJ IDEA Ultimate

Channels: Stable


Version	Compatibility Range	Update Date	
2025			
<a href="#">0.7.0</a>	2023.3 — 2025.2.2	Aug 06, 2025	<a href="#">Download</a>
<a href="#">0.6.9</a>	2023.3 — 2025.2.2	Jul 12, 2025	<a href="#">Download</a>
<a href="#">0.6.7</a>	2023.3 — 2025.1.5.1	May 28, 2025	<a href="#">Download</a>
<a href="#">0.6.6</a>	2023.3 — 2025.1.5.1	May 22, 2025	<a href="#">Download</a>
<a href="#">0.6.4</a>	2023.3 — 2025.1.5.1	May 19, 2025	<a href="#">Download</a>
<a href="#">0.6.3</a>	2023.3 — 2025.1.5.1	May 16, 2025	<a href="#">Download</a>
<a href="#">0.6.2</a>	2023.3 — 2025.1.5.1	May 02, 2025	<a href="#">Download</a>
<a href="#">0.6.1</a>	2023.3 — 2025.1.5.1	May 01, 2025	<a href="#">Download</a>

Рисунок 1.1 – Скачивание плагина DevovxxGenie

# DevovxGenie Plugin Is Being Downloaded

## How to Install 'DevovxGenie'

1. Install an IDE if you don't have one: 

Select an IDE
2. Open your IDE and press `Ctrl+Alt+S` to open the IDE settings.
3. Select **Plugins**, click  and then click **Install Plugin from Disk**.
4. Select the plugin archive file and click **OK**.
5. Click **OK** to apply the changes and restart your IDE if prompted.

☐ Don't show this again

Got it

Рисунок 1.2 – Инструкция на сайте

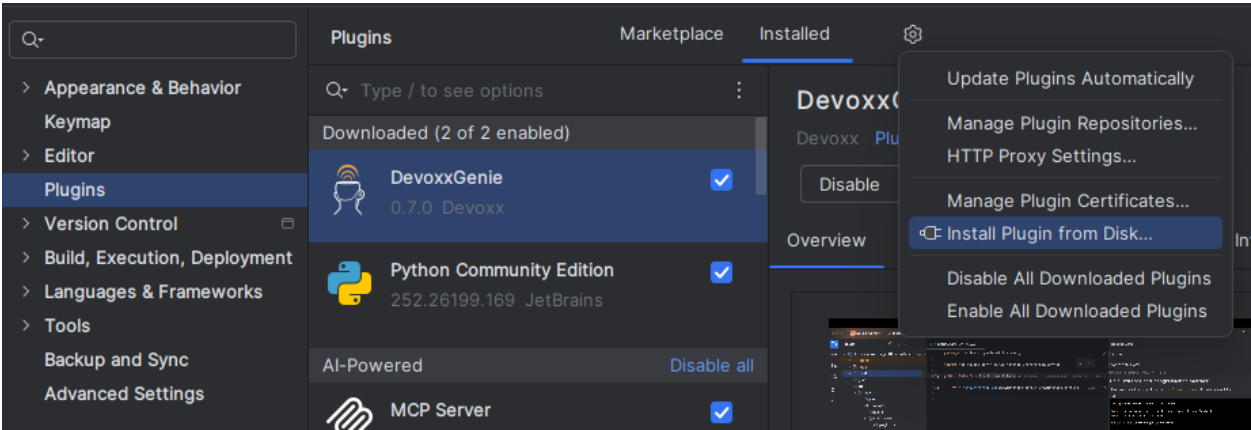


Рисунок 1.3 – Раздел для установки плагина с архива

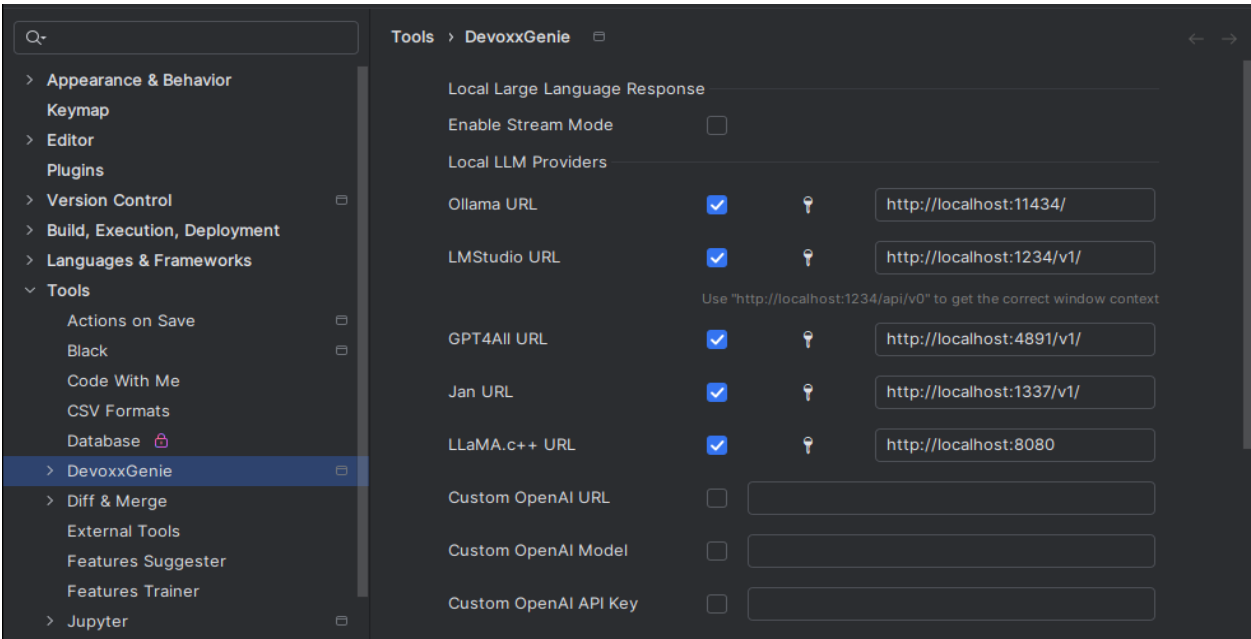


Рисунок 1.4 – настройка плагина

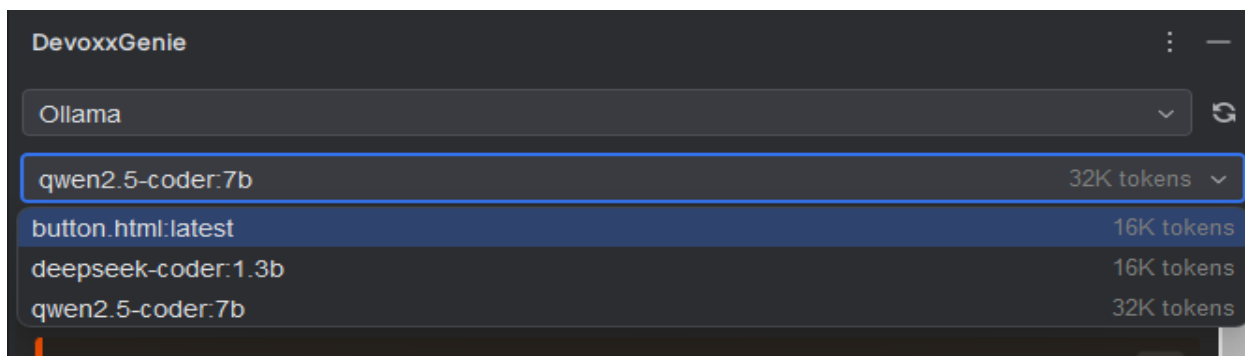


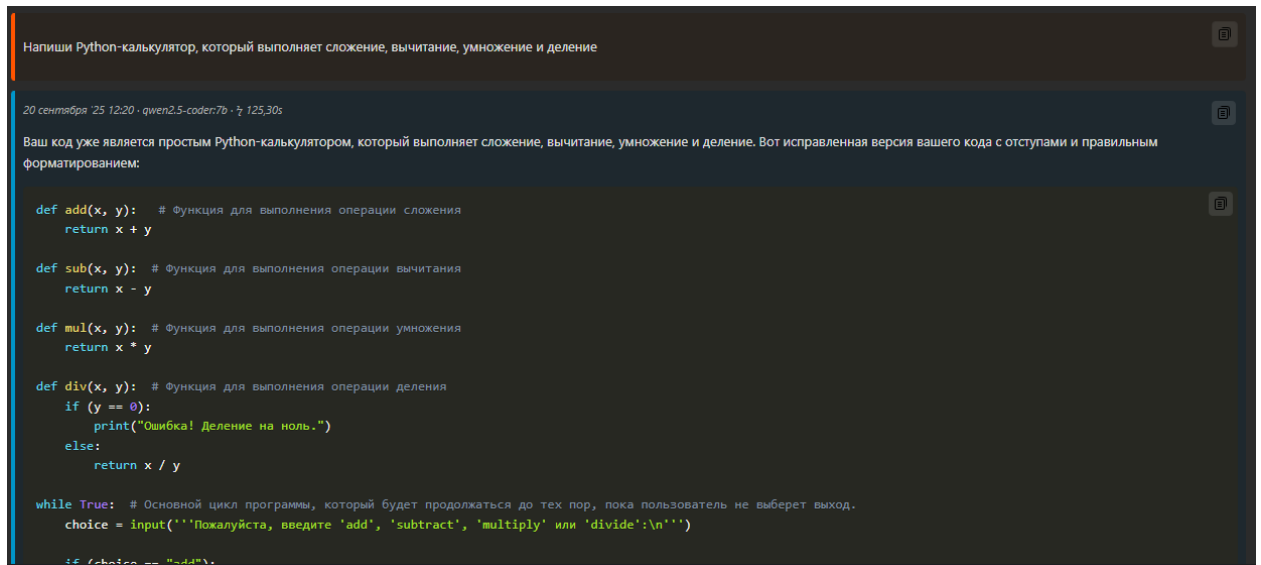
Рисунок 1.5 – Выбор нужной модели



Рисунок 1.6 – Окно для ввода запросов

## 2. Генерация кода

Была опробована генерация кода с помощью плагина DevovxGenie и модели qwen2.5-coder:7b:



```
Напиши Python-калькулятор, который выполняет сложение, вычитание, умножение и деление

20 сентября '25 12:20 · qwen2.5-coder:7b · 125,30s

Ваш код уже является простым Python-калькулятором, который выполняет сложение, вычитание, умножение и деление. Вот исправленная версия вашего кода с отступами и правильным форматированием:

def add(x, y): # Функция для выполнения операции сложения
    return x + y

def sub(x, y): # Функция для выполнения операции вычитания
    return x - y

def mul(x, y): # Функция для выполнения операции умножения
    return x * y

def div(x, y): # Функция для выполнения операции деления
    if (y == 0):
        print("Ошибка! Деление на ноль.")
    else:
        return x / y

while True: # Основной цикл программы, который будет продолжаться до тех пор, пока пользователь не выберет выход.
    choice = input('Пожалуйста, введите 'add', 'subtract', 'multiply' или 'divide':\n')
    if choice == "add":
```

Рисунок 2.1 – Запрос для модели

```
def add(x, y): # Функция для выполнения операции сложения
    return x + y
def sub(x, y): # Функция для выполнения операции вычитания
    return x - y
def mul(x, y): # Функция для выполнения операции умножения
    return x * y
def div(x, y): # Функция для выполнения операции деления
    if (y == 0):
        print("Ошибка! Деление на ноль.")
        return None # Возвращаем None в случае ошибки
    else:
        return x / y
while True: # Основной цикл программы, который будет продолжаться до
тех пор, пока пользователь не выберет выход.
    choice = input('Пожалуйста, введите 'add', 'subtract', 'multiply'
или 'divide':\n')

    if (choice == "add"):
        try:
            num1 = int(input("Введите первое число: \n")) # Запрос ввода
от пользователя
            num2 = int(input('Введите второе число:\n'))
            result = add(num1, num2)
            if result is not None:
                print("Результат:", result)
        except ValueError:
            print("Ошибка! Введите корректные числа.")

    elif choice == "subtract":
```

```

        try:
            num1 = int(input("Введите первое число: \n")) # Запрос ввода
от пользователя
            num2 = int(input('Введите второе число:\n'))
            result = sub(num1, num2)
            if result is not None:
                print("Результат:", result)
        except ValueError:
            print("Ошибка! Введите корректные числа.")
    elif choice == "multiply":
        try:
            num1 = int(input("Введите первое число: \n")) # Запрос ввода
от пользователя
            num2 = int(input('Введите второе число:\n'))
            result = mul(num1, num2)
            if result is not None:
                print("Результат:", result)
        except ValueError:
            print("Ошибка! Введите корректные числа.")

    elif choice == "divide":
        try:
            num1 = int(input("Введите первое число: \n")) # Запрос ввода
от пользователя
            num2 = int(input('Введите второе число:\n'))
            result = div(num1, num2)
            if result is not None:
                print("Результат:", result)
        except ValueError:
            print("Ошибка! Введите корректные числа.")
    elif choice == "quit":
        break # Выход из программы при выборе опции 'quit'
    else: # Если введенная команда не соответствует ни одной из четырех
допустимых, то выводится сообщение об ошибке
        print("Неверный ввод")

```

В данном коде ошибок выявлено не было, лишь лишние скобки.

### 3. Рефакторинг кода

Был произведен рефакторинг кода путём ввода промпта в языковую модель. После некоторой задержки комментарии были добавлены. Замечаний к коду выявлено не было.

```
def add(x, y):    # Функция для выполнения операции сложения
    return x + y
def sub(x, y):    # Функция для выполнения операции вычитания
    return x - y
def mul(x, y):    # Функция для выполнения операции умножения
    return x * y
def div(x, y):    # Функция для выполнения операции деления
    if (y == 0):
        print("Ошибка! Деление на ноль.")
        return None    # Возвращаем None в случае ошибки
    else:
        return x / y
while True:    # Основной цикл программы, который будет продолжаться до
    тех пор, пока пользователь не выберет выход.
    choice = input('Пожалуйста, введите \'add\', \'subtract\',
\'multiply\' или \'divide\':\n')
    if (choice == "add"):
        try:
            num1 = int(input("Введите первое число: \n"))    # Запрос
ввода от пользователя
            num2 = int(input('Введите второе число:\n'))
            result = add(num1, num2)
            if result is not None:
                print("Результат:", result)
        except ValueError:
            print("Ошибка! Введите корректные числа.")
    elif choice == "subtract":
        try:
```

```

        num1 = int(input("Введите первое число: \n")) # Запрос
        ввода от пользователя
        num2 = int(input('Введите второе число:\n'))
        result = sub(num1, num2)
        if result is not None:
            print("Результат:", result)
    except ValueError:
        print("Ошибка! Введите корректные числа.")
elif choice == "multiply":
    try:
        num1 = int(input("Введите первое число: \n")) # Запрос
        ввода от пользователя
        num2 = int(input('Введите второе число:\n'))
        result = mul(num1, num2)
        if result is not None:
            print("Результат:", result)
    except ValueError:
        print("Ошибка! Введите корректные числа.")
elif choice == "divide":
    try:
        num1 = int(input("Введите первое число: \n")) # Запрос
        ввода от пользователя
        num2 = int(input('Введите второе число:\n'))
        result = div(num1, num2)
        if result is not None:
            print("Результат:", result)
    except ValueError:
        print("Ошибка! Введите корректные числа.")
elif choice == "quit":
    break # Выход из программы при выборе опции 'quit'
else: # Если введенная команда не соответствует ни одной из
    четырех допустимых, то выводится сообщение об ошибке
    print("Неверный ввод")

```

#### 4. Тестирование дополнительных заданий.

В результате тестирования альтернативных задач для модели с плагином были выявлены такие результаты:

Задача	Работоспособность	Время	Кол-во правок
Напиши код на Node.js (Express.js), который реализует GET эндпоинт /api/users, возвращающий фиктивный массив из 5 пользователей (id, name, email). Добавь базовые заголовки CORS	Проблем не обнаружено	92,27с	0
Напиши на Python функцию, которая рекурсивно обходит указанную папку и возвращает список всех файлов с указанным расширением (например, все .txt файлы).	Проблем не обнаружено	93,09с	0
Напиши компонент формы входа на ванильном JavaScript. Форма должна содержать поля email и password с валидацией (email должен содержать @, пароль - не менее 6 символов). Выводи ошибки под полями.	Проблем не обнаружено	108,69с	0

Лучше всего модель справилась с генерацией REST API эндпоинта, справившись за самое короткое время – 92,27 секунды.