

## Отчет по лабораторной работе № 2

Тема: AI-powered редакторы кода

Цели работы

- Освоить работу с современными AI-powered редакторами кода
- Сравнить возможности различных AI-ассистентов для программирования
- Научиться эффективно формулировать запросы для генерации кода
- Изучить лучшие практики совместной работы с AI при разработке
- Оценить качество и эффективность AI-генерируемого кода

Были выбраны 2 редактора кода

Cursor:

- Основан на VS Code с интегрированным AI
- GPT-4, Claude интеграция
- Контекстное понимание всего проекта
- Функция "Composer" для больших изменений

Windsurf (Codeium):

- Бесплатный AI-ассистент
- Поддержка 70+ языков программирования
- Автодополнение в реальном времени
- Чат-интерфейс для объяснения кода

## 1. Установка и настройка редакторов

### 1.1 Установка Windsurf

Для регистрации необходим VPN

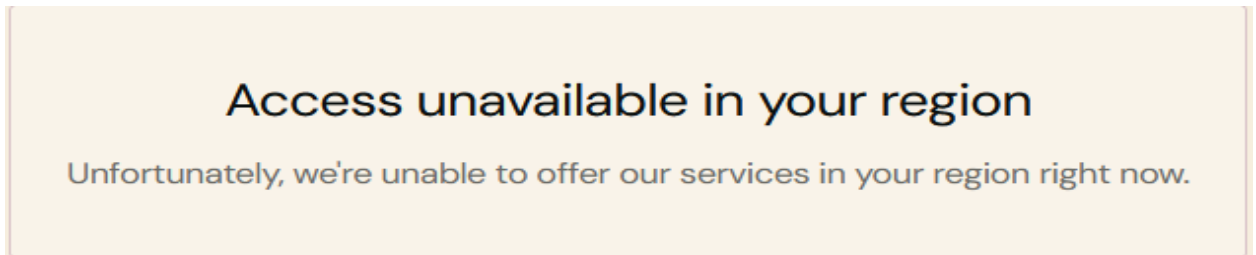


Рисунок 1.1 – Не доступен в вашем регионе

Устанавливается как плагин для VS Code, после понадобится авторизоваться через сайт



Рисунок 1.2 – Плагин в VS Code

### 1.2 Установка Cursor

После скачивания с официального сайта запустить установщик

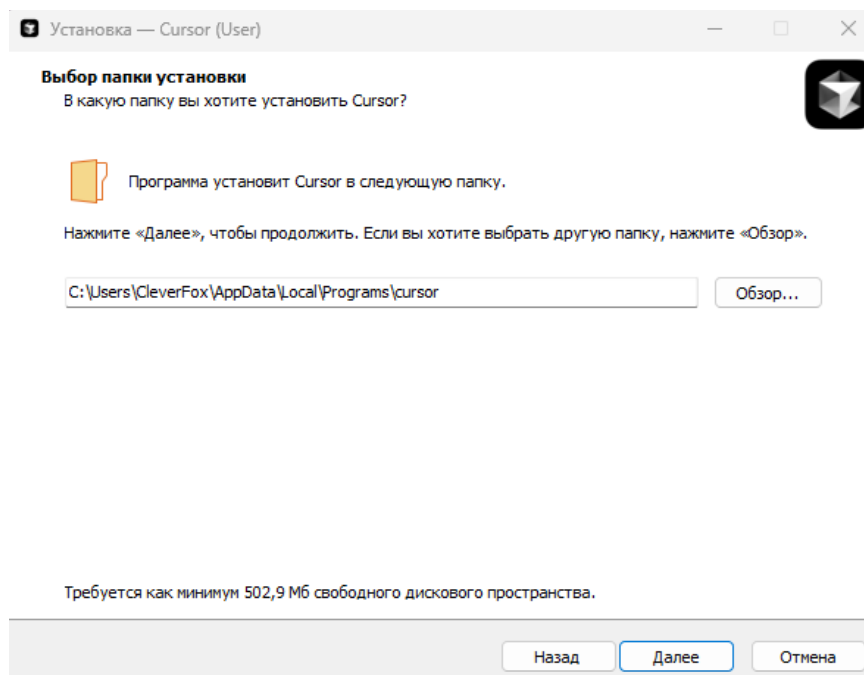


Рисунок 1.3 – Выбор директории установки

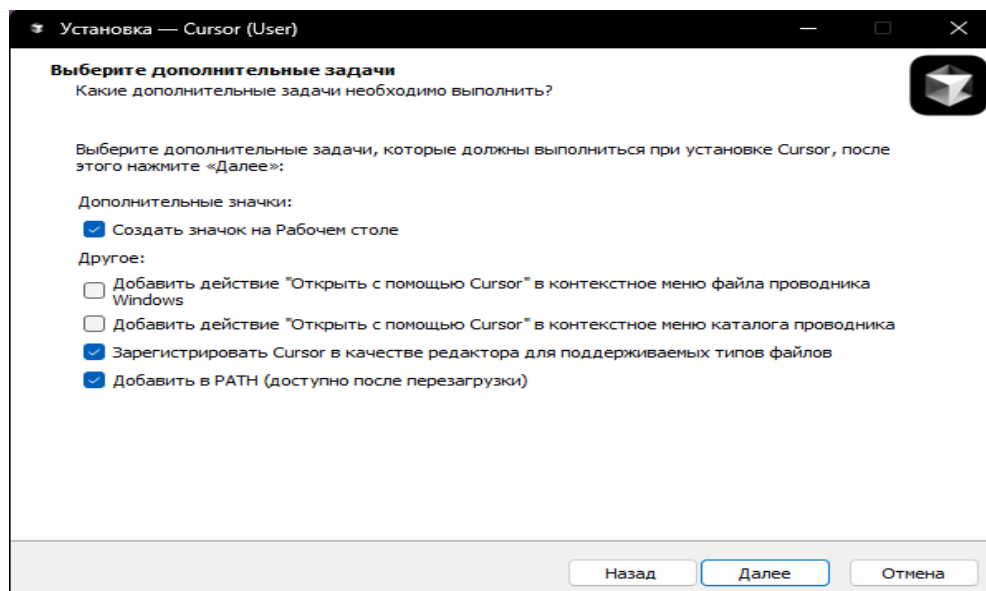


Рисунок 1.4 – Дополнительные задачи

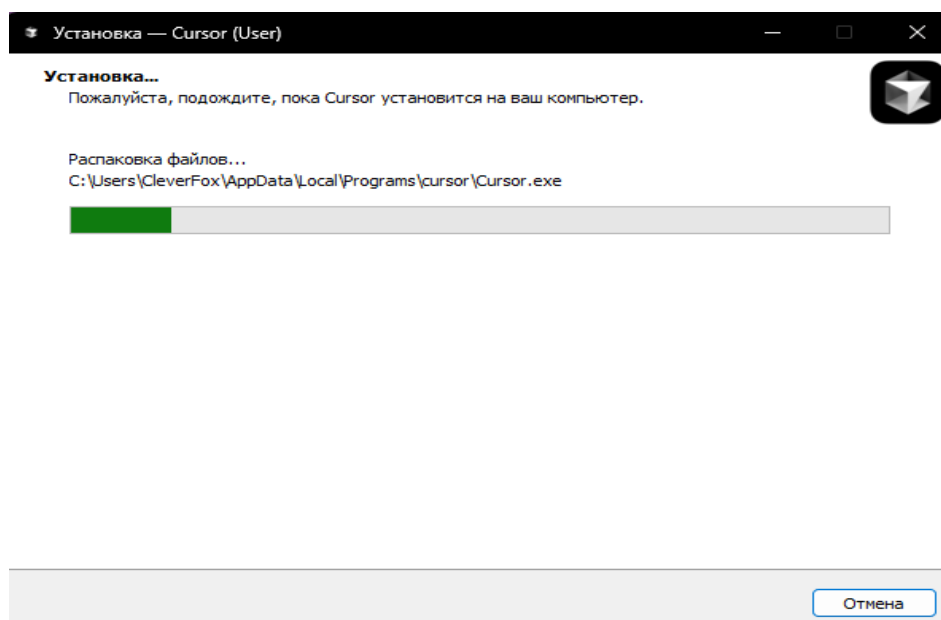


Рисунок 1.5 – Процесс установки

Далее пользователю предлагается выбрать внешний вид редактора.

В дальнейшем при использовании и отправки запросов будет предлагаться установка дополнительных расширений в виде небольших всплывающих окон.

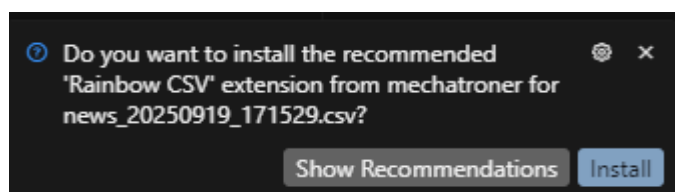


Рисунок 1.6 – Всплывающее окно рекомендаций по установке

## 2. Реализация заданий

### Задание 1: Калькулятор с GUI

Техническое задание: Создать калькулятор с графическим интерфейсом на Python (tkinter или PyQt).

Функциональность:

- Основные операции: +, -, \*, /
- Операции с памятью: M+, M-, MR, MC
- Обработка ошибок деления на ноль
- История последних 5 вычислений

Windsurf (Codeium):

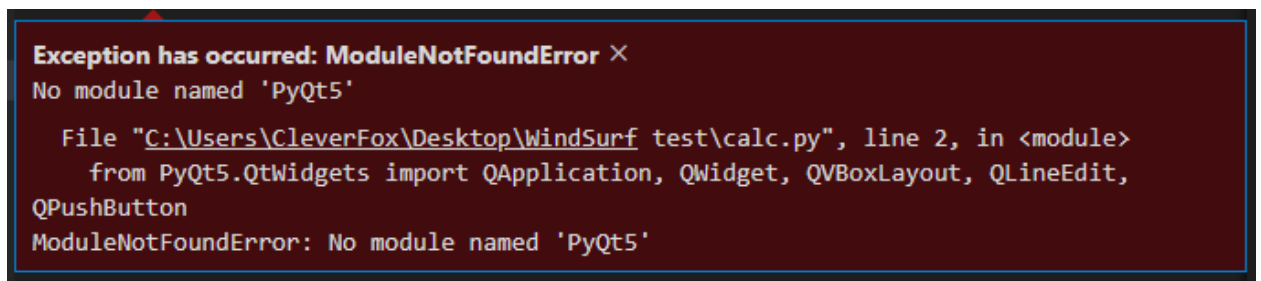


Рисунок 2.1 – Ошибка, нет модуля с названием «PyQt5»

Cursor:

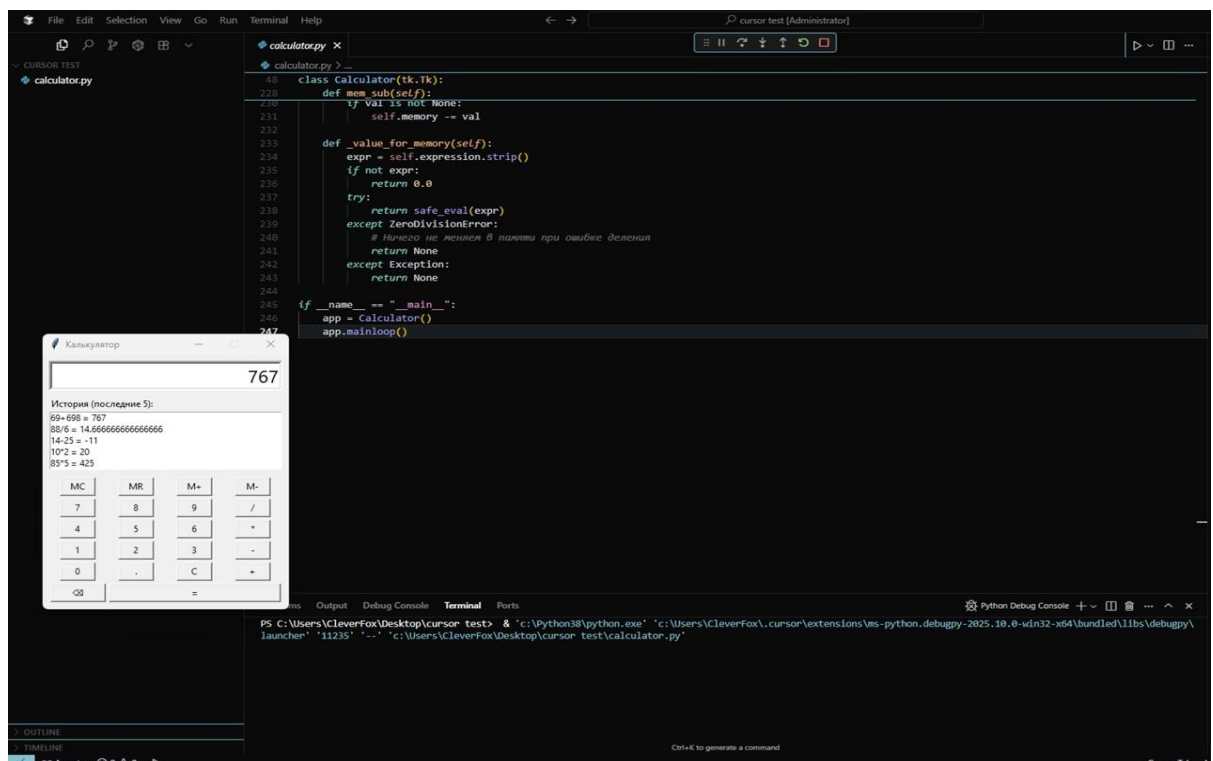


Рисунок 2.2 – Калькулятор с GUI

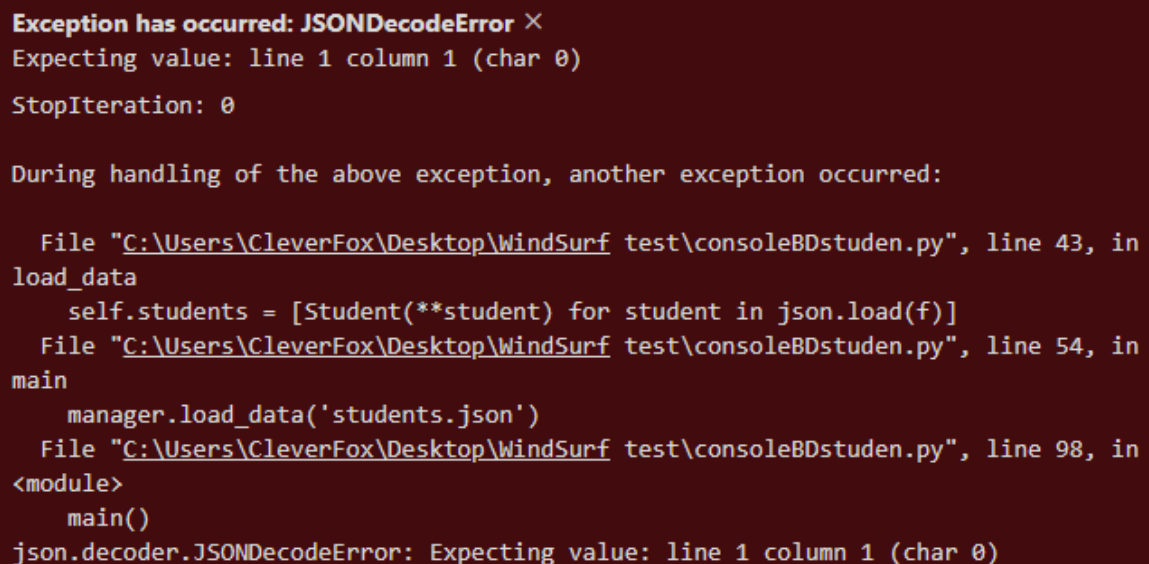
## Задание 2: Система управления студентами

Техническое задание: Консольное приложение для управления базой данных студентов.

Функциональность:

- Добавление/удаление/редактирование студентов
- Поиск по имени, группе, номеру зачетки
- Подсчет среднего балла
- Сохранение данных в JSON файл
- Валидация входных данных

Windsurf (Codeium):



```
Exception has occurred: JSONDecodeError ×
Expecting value: line 1 column 1 (char 0)
StopIteration: 0

During handling of the above exception, another exception occurred:

File "C:\Users\CleverFox\Desktop\WindSurf test\consoleBDstuden.py", line 43, in
load_data
    self.students = [Student(**student) for student in json.load(f)]
File "C:\Users\CleverFox\Desktop\WindSurf test\consoleBDstuden.py", line 54, in
main
    manager.load_data('students.json')
File "C:\Users\CleverFox\Desktop\WindSurf test\consoleBDstuden.py", line 98, in
<module>
    main()
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
```

Рисунок 2.3 – Ошибка

Cursor:

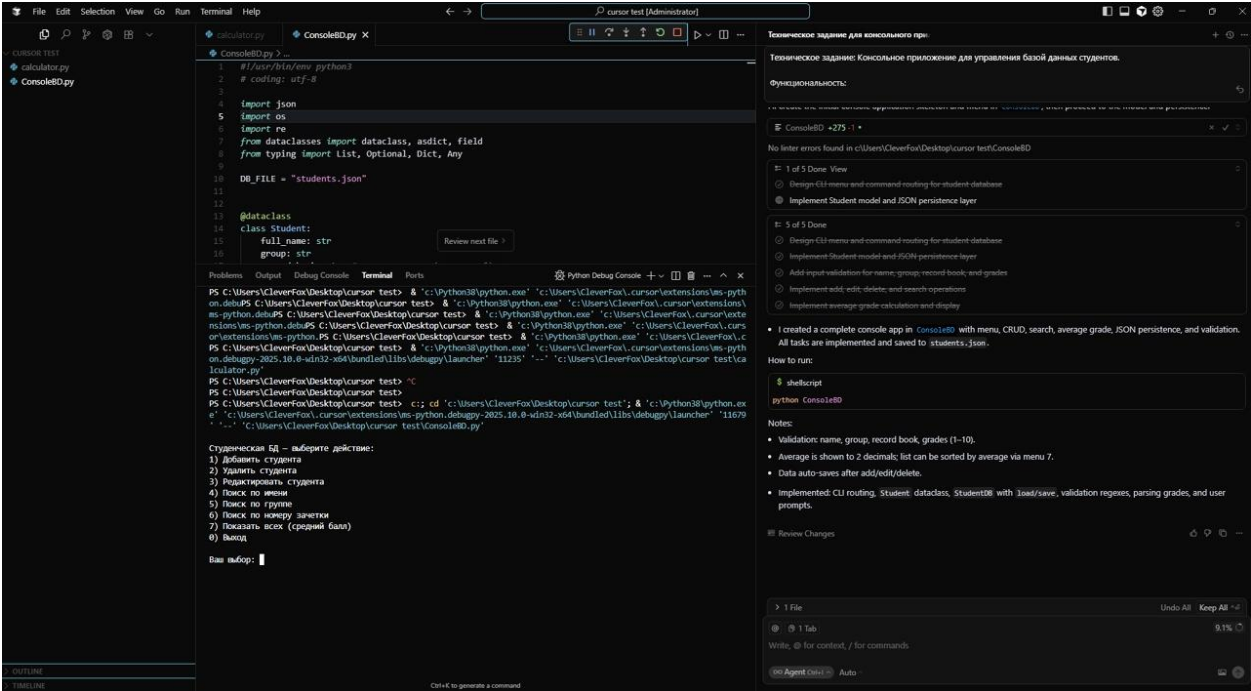


Рисунок 2.4 – Система управления студентами

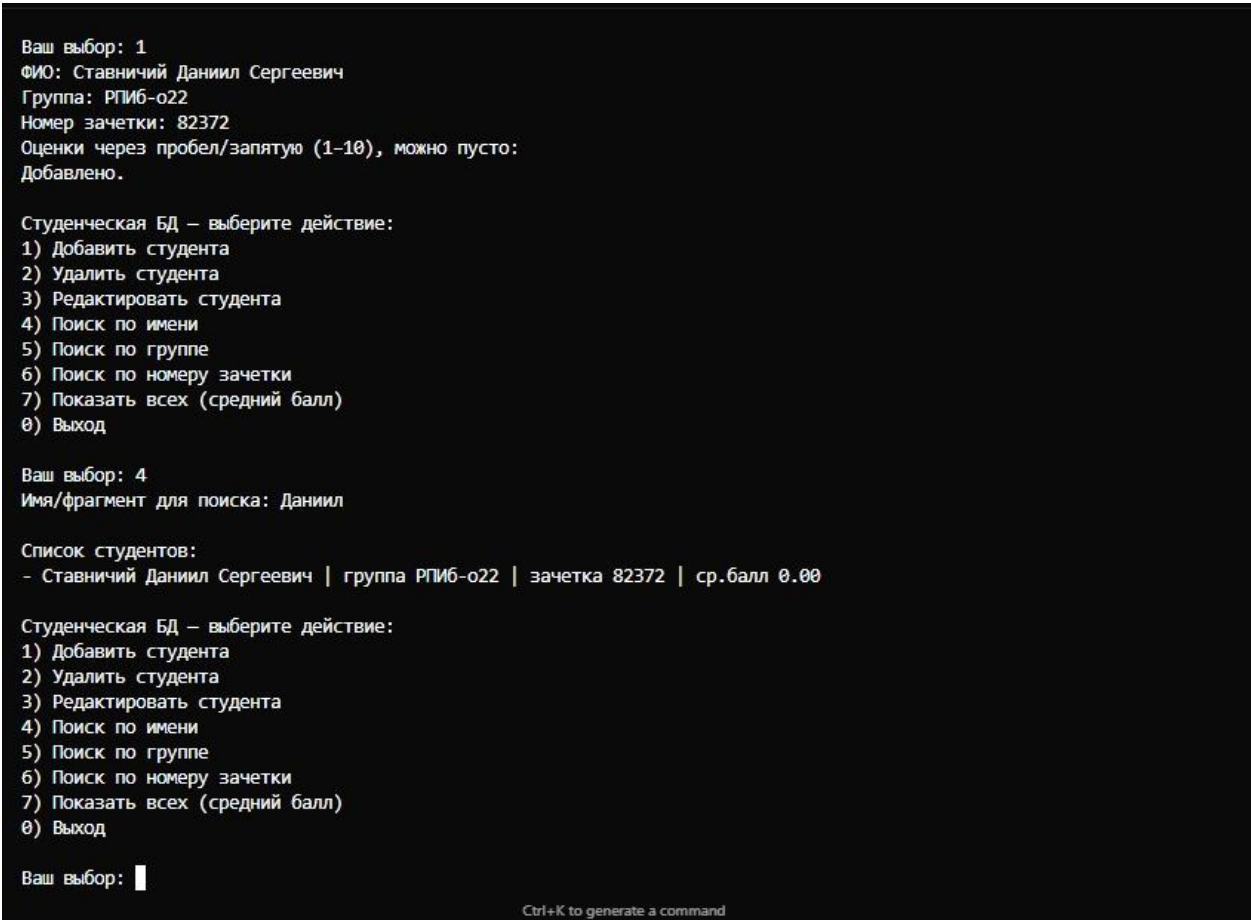


Рисунок 2.5 – Добавление, поиск

Ваш выбор: 2  
Номер зачетки для удаления: 82372  
Удалено.

Студенческая БД – выберите действие:

- 1) Добавить студента
- 2) Удалить студента
- 3) Редактировать студента
- 4) Поиск по имени
- 5) Поиск по группе
- 6) Поиск по номеру зачетки
- 7) Показать всех (средний балл)
- 0) Выход

Ваш выбор: 4  
Имя/фрагмент для поиска: Даниил  
Нет записей.

Студенческая БД – выберите действие:

- 1) Добавить студента
- 2) Удалить студента
- 3) Редактировать студента
- 4) Поиск по имени
- 5) Поиск по группе
- 6) Поиск по номеру зачетки
- 7) Показать всех (средний балл)
- 0) Выход

Ваш выбор: █

Ctrl+K to generate a command

Рисунок 2.6 – Удаление

Список студентов:

- Серёгин Алексей | группа РПИб-о22 | зачетка 12323 | ср.балл 4.50
- Ставничий Даниил Сергеевич | группа РПИб-о22 | зачетка 82372 | ср.балл 4.33

Студенческая БД – выберите действие:

- 1) Добавить студента
- 2) Удалить студента
- 3) Редактировать студента
- 4) Поиск по имени
- 5) Поиск по группе
- 6) Поиск по номеру зачетки
- 7) Показать всех (средний балл)
- 0) Выход

Ваш выбор: █

Рисунок 2.7 – Вычисление среднего балла

Ваш выбор: 3

Номер зачетки для редактирования: 82372

Нажмите Enter, чтобы оставить без изменений.

ФИО [Ставничий Даниил Сергеевич]:

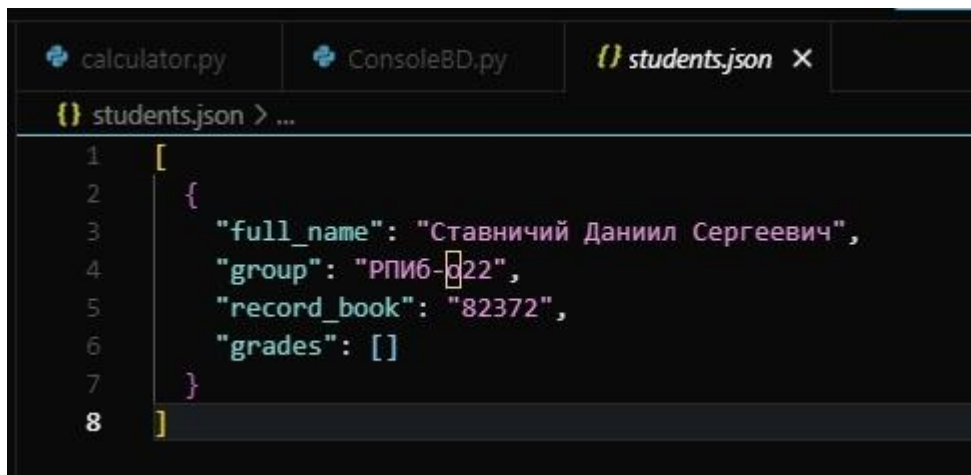
Группа [РПИб-о22]:

Оценки (через пробел/запятую) [оставить – Enter]: ф в в в в

Некорректные оценки – пропускаю изменение.

Обновлено.

Рисунок 2.8 – Валидация



```
calculator.py ConsoleBD.py students.json x
students.json > ...
1  [
2    {
3      "full_name": "Ставничий Даниил Сергеевич",
4      "group": "РПИ6-022",
5      "record_book": "82372",
6      "grades": []
7    }
8  ]
```

Рисунок 2.9 – Сохранение в файле

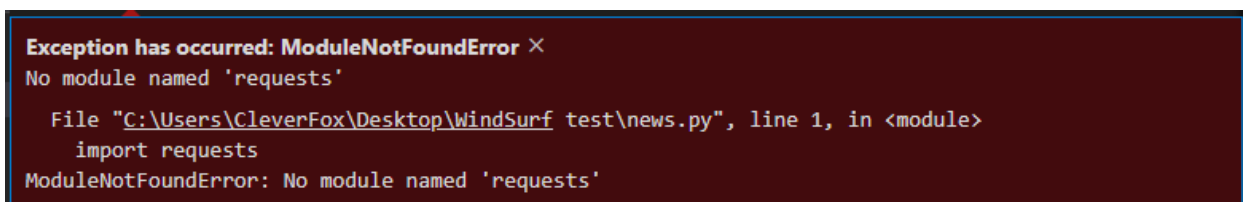
### Задание 3: Веб-скрейпер новостей

Техническое задание: Программа для сбора и анализа новостей с сайтов.

Функциональность:

- Парсинг заголовков с 2-3 новостных сайтов
- Фильтрация по ключевым словам
- Сохранение в CSV с датой/временем
- Базовая статистика (частота слов)
- Обработка ошибок сети

Windsurf (Codeium):



```
Exception has occurred: ModuleNotFoundError x
No module named 'requests'

File "C:\Users\CleverFox\Desktop\WindSurf test\news.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
```

Рисунок 2.10 – Ошибка модуля



Cursor:

```
Источник: BBC World
URL: https://feeds.bbc.co.uk/news/world/rss.xml
[!] Ошибка источника: Network error: curl error _ssl.c:1091: The handshake operation timed out

Источник: NYTimes World
URL: https://rss.nytimes.com/services/xml/rss/nyt/World.xml
Получено элементов: 59
После фильтра: 59

Источник: The Guardian World
URL: https://www.theguardian.com/world/rss
Получено элементов: 45
После фильтра: 45

Сохранено в файл: news_20250919_171529.csv

Топ-20 слов (без стоп-слов, длина ≥ 3):
trump: 18
new: 8
says: 7
state: 6
castle: 5
europe: 5
live: 5
people: 5
afghanistan: 4
british: 4
china: 4
deal: 4
gaza: 4
has: 4
more: 4
palestinian: 4
windsor: 4
alleged: 3
amid: 3
between: 3
PS C:\Users\CleverFox\Desktop\cursor test> |
```

Рисунок 2.11 – Работа программы



```
1 source,title,link,published,fetched_at
2 NYTimes World,"Battered but Undefeated, Hamas Remains a Fighting Force in Gaza",https://www.nytimes.com/2025/09/19/world/middleeast/hamas-remaining-in-g
3 NYTimes World,Autocrats Move Quickly to Fill Void as Trump Retreats From U.N.,https://www.nytimes.com/2025/09/19/world/asia/trump-un-china-authoritarian
4 NYTimes World,How Kenyan Villagers Saved Their Sacred Caves From a Mining Company,https://www.nytimes.com/2025/09/19/world/africa/kenya-mining-caves.htm
5 NYTimes World,"Still Divided, a Syrian City Ruined in War Edges Back to Life",https://www.nytimes.com/2025/09/19/world/middleeast/syria-deir-al-zour.htm
6 NYTimes World,"Flattery and Flourish for Trump. For His U.K. Hosts, Relief at Avoiding Strife.",https://www.nytimes.com/2025/09/19/world/europe/trump-ki
7 NYTimes World,How Xi Is Using a TikTok Tradeoff to Court Trump,https://www.nytimes.com/2025/09/19/world/asia/tiktok-xi-trump-china.html,"Fri, 19 Sep 202
8 NYTimes World,The Plot to Free the Nuns,https://www.nytimes.com/2025/09/19/world/europe/nuns-austria-sisters.html,"Fri, 19 Sep 2025 11:55:11 +0000",2025
9 NYTimes World,Taliban Rebuffs Trump's Effort to Regain Air Base in Afghanistan,https://www.nytimes.com/2025/09/19/world/asia/trump-bagram-taliban.html,"I
10 NYTimes World,Europe Announces New Sanctions to Ramp Up Pressure on Russia,https://www.nytimes.com/2025/09/19/world/europe/eu-sanctions-russia-ukraine.h
11 NYTimes World,"Harrowing Beyond Description": Dozens Killed While at Prayer in Sudan,https://www.nytimes.com/2025/09/19/world/africa/sudan-strike-mosque
12 NYTimes World,"Taliban Releases British Couple, Peter and Barbie Reynolds, Detained for Months in Afghanistan",https://www.nytimes.com/2025/09/19/world/a
13 NYTimes World,How China Stands to Gain as the U.S. Steps Away From the U.N.,https://www.nytimes.com/2025/09/19/world/asia/china-united-nations-trump.htm
14 NYTimes World,The Road,https://www.nytimes.com/2025/09/18/world/middleeast/gaza-city-evacuation.html,"Thu, 18 Sep 2025 09:56:11 +0000",2025-09-19T17:15:
15 NYTimes World,The Hottest New Defense Against Drones? Lasers,https://www.nytimes.com/2025/09/18/world/europe/drones-laser-weapons.html,"Thu, 18 Sep 2025
16 NYTimes World,She's Starting a Blood Bank for Zoos,https://www.nytimes.com/2025/09/19/science/zoo-veterinarian-blood.html,"Fri, 19 Sep 2025 09:00:40 +000
17 NYTimes World,Drones vs. Lasers,https://www.nytimes.com/2025/09/19/briefing/nato-drones-russia.html,"Fri, 19 Sep 2025 14:10:06 +0000",2025-09-19T17:15:2
18 NYTimes World,West Africans Deported by the U.S. Sue Ghana for Rights Violations,https://www.nytimes.com/2025/09/18/world/africa/us-deportees-sue-ghana.l
19 NYTimes World,Where Will Gazans Go?,https://www.nytimes.com/video/world/middleeast/100000010405069/where-will-gazans-go.html,"Thu, 18 Sep 2025 21:54:13
20 NYTimes World,Drones vs. Lasers,https://www.nytimes.com/2025/09/18/briefing/nato-air-defense-trump-media-britain-tech-deal.html,"Thu, 18 Sep 2025 21:22:
21 NYTimes World,"Trump and Starmer, Skirting Differences, Praise Each Other in U.K.",https://www.nytimes.com/2025/09/18/world/europe/trump-starmer-britain
22 NYTimes World,Exxon Urges Europe to Repeal Rules to Make Companies Track Climate Pollution,https://www.nytimes.com/2025/09/18/climate/exxon-woods-europe
23 NYTimes World,"José Maria Tojeira, 78, Dies; Pursued Killers of His Fellow Priests",https://www.nytimes.com/2025/09/18/world/americas/jose-maria-tojeira
24 NYTimes World,Climate Activists Argue 3 Executive Orders Violate Their Constitutional Rights,https://www.nytimes.com/2025/09/18/climate/ligthtiser-trial
25 NYTimes World,U.S. to Cut Out Nongovernmental Groups in Refocusing Health Aid,https://www.nytimes.com/2025/09/18/us/politics/us-aid-health-ngos.html,"Thu
26 NYTimes World,E.P.A. Keeps Polluters on the Hook to Clean Up 'Forever Chemicals',https://www.nytimes.com/2025/09/18/climate/epa-pfas-forever-chemicals.h
27 NYTimes World,"Wildfire Smoke Will Kill Thousands More by 2050, Study Finds",https://www.nytimes.com/2025/09/18/climate/wildfire-smoke-health-deaths.htm
28 NYTimes World,"Do Reptiles Have Moods, Too?",https://www.nytimes.com/2025/09/18/science/reptiles-cognition-moods.html,"Thu, 18 Sep 2025 14:41:53 +0000",
29 NYTimes World,Trump Administration Reinstates More Difficult Citizenship Test,https://www.nytimes.com/2025/09/18/world/europe/us-citizenship-test.html,"
30 NYTimes World,Assailant Kills 2 Israeli Soldiers at West Bank-Jordan Border,https://www.nytimes.com/2025/09/18/world/middleeast/allenby-shooting-jordan-1
```

Рисунок 2.12 – Файл csv

### 3. Сравнительное тестирование

Оценка от 1 до 5 где: 1 это самая низкая оценка, 5 самая высокая.

| Критерий               | Cursor  | Windsurf (Codeium) |
|------------------------|---------|--------------------|
| Скорость генерации     | Высокая | Очень высокая      |
| Качество кода          | 5       | 3                  |
| Понимание контекста    | 5       | 4                  |
| Объяснение решений     | 4       | 5                  |
| Исправление ошибок     | 5       | 2                  |
| Интерфейс и интеграция | 5       | 4                  |

В ходе сравнения был сделан вывод.

Cursor подходит для:

- более сложных проектов
- Рефакторинга
- Работы с архитектурой

Windsurf подходит для:

- повседневного кодирования
- бесплатного использования
- как дополнение к основному инструменту

## 4.Специальные тесты

### 4.1 Тест на объяснение кода:

Промпт: "Explain this quicksort implementation step by step, including how partitioning works."

```
def quicksort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quicksort(arr, low, pi - 1)
        quicksort(arr, pi + 1, high)

def partition(arr, low, high):
    pivot = arr[high]
    i = low - 1
    for j in range(low, high):
        if arr[j] <= pivot:
            i += 1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    return i + 1
```

#### Windsurf (Codeium):

Объяснение в разговорной манере с несколькими однотипными примерами

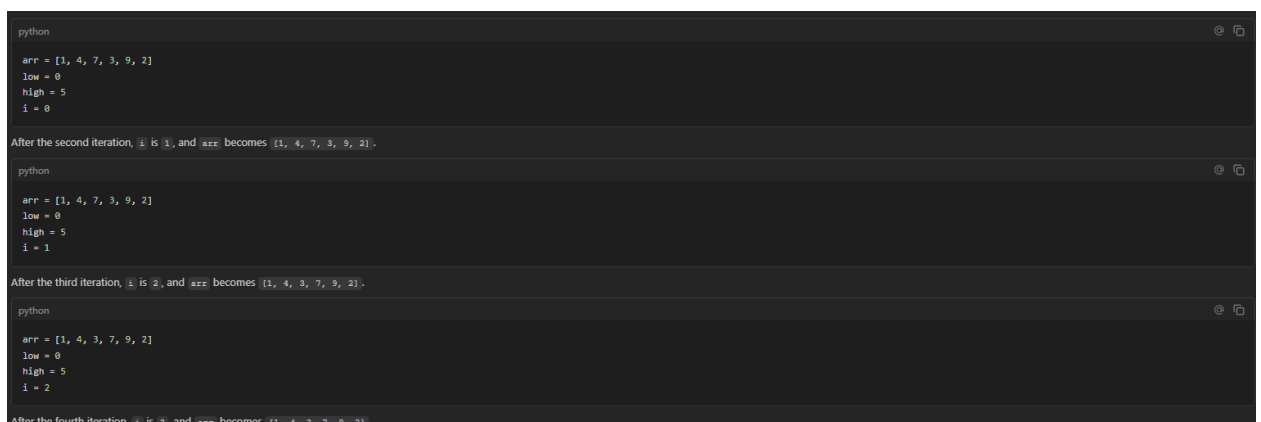


Рисунок 4.1. Примеры Windsurf

## Cursor:

Краткие разъяснения с небольшим примером

### Quicksort flow

1. Check base case: if `low >= high`, subarray has 0–1 elements and is already sorted.
2. Partition: `pi = partition(arr, low, high)` puts the pivot in its correct place `pi` and ensures:
  - `arr[low..pi-1] ≤ pivot`
  - `arr[pi+1..high] > pivot`
3. Recurse on left part: `quicksort(arr, low, pi - 1)`
4. Recurse on right part: `quicksort(arr, pi + 1, high)`

### Partition (Lomuto scheme) step-by-step

- Pivot: `pivot = arr[high]` (last element).
- Boundary index: `i = low - 1` (end of the "`≤ pivot`" region).
- Iterate `j` from `low` to `high - 1`:
  - If `arr[j] ≤ pivot`:
    - Expand the "`≤ pivot`" region by incrementing `i`.
    - Swap `arr[i]` and `arr[j]` to place this element into the left region.
- After the loop, place the pivot right after the "`≤ pivot`" region:
  - Swap `arr[i + 1]` with `arr[high]`.
- Return `i + 1` as the pivot's final index.

### Mini example

Given `arr = [3, 7, 2, 5]`, `low=0`, `high=3`:

- Pivot = 5, `i = -1`
- `j=0`: `3 ≤ 5 → i=0`, swap `arr[0]`, `arr[0] → [3,7,2,5]`
- `j=1`: `7 ≤ 5?` no
- `j=2`: `2 ≤ 5 → i=1`, swap `arr[1]`, `arr[2] → [3,2,7,5]`
- Final swap pivot with `arr[i+1]=arr[2]`: `→ [3,2,5,7]`, return 2

Then quicksort left `[3,2]` and right `[7]`.

Рисунок 4.2. Примеры Cursor

## 4.2 Тест на рефакторинг

Промпт: "Refactor this code to use better practices: enum for operations, proper error handling, type hints, and documentation."

```
def calc(x, y, op):
```

```

if op == 1:
    return x + y
elif op == 2:
    return x - y
elif op == 3:
    return x * y
elif op == 4:
    if y != 0:
        return x / y
    else:
        return "Error"
else:
    return "Invalid operation"

```

### **Windsurf (Codeium):**

```

from enum import Enum

```

```

class Operation(Enum):

```

```

    ADD = 1

```

```

    SUBTRACT = 2

```

```

    MULTIPLY = 3

```

```

    DIVIDE = 4

```

```

def calc(x: float, y: float, op: Operation) -> Union[float, str]:
    """

```

Perform a calculation based on the given operation and return the result.

Args:

x (float): The first operand.

y (float): The second operand.

op (Operation): The operation to perform.

Returns:

float or str: The result of the calculation if successful, or an error message if an error occurred.

```

    """

```

```

if op == Operation.ADD:

```

```

    return x + y

```

```

elif op == Operation.SUBTRACT:
    return x - y
elif op == Operation.MULTIPLY:
    return x * y
elif op == Operation.DIVIDE:
    if y != 0:
        return x / y
    else:
        return "Error"
else:
    return "Invalid operation"

```

### **Cursor:**

```

from enum import Enum
from typing import Union
Number = Union[int, float]
class Operation(Enum):
    ADD = 1
    SUB = 2
    MUL = 3
    DIV = 4
def calc(x: Number, y: Number, op: Operation) -> float:
    """
    Calculate a binary arithmetic operation.
    Args:
        x: Left operand.
        y: Right operand.
        op: Operation to perform (Operation.ADD/SUB/MUL/DIV).
    Returns:
        The numeric result as float (Python may return int for
        exact sums/products, but type hints use float).
    Raises:
        ZeroDivisionError: If op is DIV and y == 0.
        ValueError: If an unsupported operation is provided.
    """

```

```

"""
if op is Operation.ADD:
    return x + y
if op is Operation.SUB:
    return x - y
if op is Operation.MUL:
    return x * y
if op is Operation.DIV:
    if y == 0:
        raise ZeroDivisionError("Division by zero")
    return x / y
raise ValueError(f"Unsupported operation: {op!r}")
def parse_operation(op_code: Union[int, Operation]) -> Operation:
    """
    Helper to accept either an Operation or its integer code.
    """
    if isinstance(op_code, Operation):
        return op_code
    try:
        return Operation(op_code)
    except ValueError as exc:
        raise ValueError(f"Invalid operation code: {op_code}")
from exc
# Example:
# result = calc(6, 3, parse_operation(4)) # -> 2.0

```

### 4.3 Тест на отладку

Промпт: "This fibonacci function is very slow for large numbers. Optimize it using memoization or iterative approach."

```

def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

```

```
# Найти и исправить проблемы производительности
result = fibonacci(40)
print(result)
```

### **Windsurf (Codeium):**

```
def fibonacci(n, memo={}):
    """
    Calculate the nth Fibonacci number using memoization.
    Args:
        n (int): The index of the Fibonacci number to calculate.
        memo (dict, optional): A dictionary to store previously
        calculated Fibonacci numbers. Defaults to an empty dictionary.
    Returns:
        int: The nth Fibonacci number.
    """
    if n in memo:
        return memo[n]
    elif n <= 1:
        return n
    else:
        memo[n] = fibonacci(n-1, memo) + fibonacci(n-2, memo)
        return memo[n]
```

```
result = fibonacci(40)
print(result)
```

Alternatively, we can also use an iterative approach to calculate Fibonacci numbers:

```
def fibonacci(n):
    """
    Calculate the nth Fibonacci number using an iterative
    approach.
    Args:
        n (int): The index of the Fibonacci number to calculate.
```



Returns:

```
    int: The nth Fibonacci number.
    """
    if n <= 1:
        return n
    a, b = 0, 1
    for _ in range(n-1):
        a, b = b, a + b
    return b
```

```
result = fibonacci(40)
print(result)
```

### **Cursor:**

Memoization (drop-in, keeps recursion):

```
from functools import lru_cache
@lru_cache(maxsize=None)
def fibonacci(n: int) -> int:
    if n <= 1:
        return n
    return fibonacci(n - 1) + fibonacci(n - 2)
print(fibonacci(40))
```

Iterative (fastest and minimal memory):

```
def fibonacci(n: int) -> int:
    if n <= 1:
        return n
    a, b = 0, 1
    for _ in range(2, n + 1):
        a, b = b, a + b
    return b
print(fibonacci(40))
```

Optional: memoization without decorator:

```
def fibonacci(n: int, memo: dict[int, int] | None = None) -> int:
    if memo is None:
        memo = {0: 0, 1: 1}
```

```
if n in memo:
    return memo[n]
memo[n] = fibonacci(n - 1, memo) + fibonacci(n - 2, memo)
return memo[n]
```

```
print(fibonacci(40))
```

Notes:

Both approaches reduce time from exponential to linear  $O(n)$ .

Iterative is preferred for speed and stack-safety.

## 5. Эксперименты с промптами

### Эксперимент 1: Стил ь кодирования

С указанием стиля: "Write a Python function to validate email addresses using regex. Follow PEP 8 style guide, add type hints, comprehensive docstrings, and unit tests."

Cursor:

```
#!/usr/bin/env python3
# coding: utf-8
import unittest
from email_utils import is_valid_email
class TestEmailValidation(unittest.TestCase):
    def test_valid_emails(self):
        valid = [
            "simple@example.com",
            "very.common@example.com",
            "disposable.style.email.with+symbol@example.com",
            "other.email-with-hyphen@example.co.uk",
            "x@example.org",
            "user_name@example.io",
            "user.name+tag+sorting@example.com",
        ]
        for e in valid:
            with self.subTest(e=e):
                self.assertTrue(is_valid_email(e))

    def test_invalid_emails(self):
        invalid = [
            "",
            "plainaddress",
            "@missinglocal.org",
            "user.@example.com",
            ".user@example.com",
```

```

    "user..dots@example.com",
    "user@example",
    "user@example.c",
    "user@-example.com",
    "user@example-.com",
    "user@example..com",
    "user@example.com-",
    "user@example.toolongtld123",
]
for e in invalid:
    with self.subTest(e=e):
        self.assertFalse(is_valid_email(e))
if __name__ == "__main__":
    unittest.main()

```

Без указания стиля: "Write a function to check if email is valid"

Cursor:

```

import re
from typing import Final

_LOCAL_PART = r"[A-Za-z0-9](?:[A-Za-z0-9._%+-]{0,62}[A-Za-z0-9])?"
_LOCAL_DOTS_SAFE = r"(?:\[A-Za-z0-9](?:[A-Za-z0-9._%+-]{0,62}[A-
Za-z0-9])?)?"
_LABEL = r"[A-Za-z0-9](?:[A-Za-z0-9-]{0,61}[A-Za-z0-9])?"
_DOMAIN = rf"{_LABEL}(?:\[A-Za-z0-9](?:[A-Za-z0-9._%+-]{0,62}[A-
Za-z0-9])?)?"
_TLD = r"[A-Za-z]{2,}"

_EMAIL_RE: Final[re.Pattern[str]] =
re.compile(rf"^{_LOCAL_PART}{_LOCAL_DOTS_SAFE}@({_DOMAIN})$")

def is_valid_email(email: str) -> bool:
    """
    Validate an email address using a compiled regular expression.
    Rules (pragmatic, not full RFC 5322):
    - Local part: letters/digits and ._%+-; no leading/trailing
    dot; no consecutive dots.

```

- Domain: dot-separated labels (letters/digits/hyphens) without leading/trailing hyphen.

- TLD: at least 2 letters.

Args:

email: Email address to validate.

Returns:

True if the email looks valid, False otherwise.

"""

```
email = email.strip()
if not email or "@" not in email:
    return False
match = _EMAIL_RE.match(email)
if not match:
    return False
domain = match.group(2)
parts = domain.rsplit(".", 1)
if len(parts) != 2:
    return False
tld = parts[1]
return bool(re.fullmatch(_TLD, tld))
```

## **Эксперимент 2: Язык запроса**

Переведён промпт с указанием стиля «Эксперимент 1»

Заметно увеличилось время обработки и выполнения запроса

## 6. Анализ качества кода

В ходе анализа были выявлены некоторые доработки:

### Todo list

1. Валидация входных данных: например, проверка формата даты (если она вводится строкой), проверка приоритета на допустимые значения (1,2,3).
2. Обработка ошибок файловой системы: что, если файл JSON поврежден или недоступен для записи?
3. Юникод-совместимость: если задачи содержат не-ASCII символы, при работе с файлом могут возникнуть проблемы.

### Валидация email

1. Полнота валидации. Например, email с кавычками или специальными символами.
2. Ложные срабатывания: некоторые несуществующие TLD могут быть приняты, так как проверяется только длина и символы.

### Рефакторинг калькулятора

1. Локализация: исключения на английском языке, в то время как приложение может быть на русском.
2. Расширяемость: если нужно добавить новую операцию, требуется изменить Enum и функцию calc.