**Тема:**  *Применение отладчиков*

**Формат:** Индивидуальный проект

**Цель:** Освоить базовые навыки работы с отладчиком для поиска и исправления ошибок в коде.  
**Инструменты:** VS Code (с расширением Python/PyCharm) / Visual Studio (для C++), встроенный отладчик.

**Тема:**  *Применение отладчиков*

**Формат:** Индивидуальный проект

**Цель:** Освоить базовые навыки работы с отладчиком для поиска и исправления ошибок в коде.  
**Инструменты:** VS Code (с расширением Python/PyCharm) / Visual Studio (для C++), встроенный отладчик.

**Теоретический минимум:**

1. **Для чего нужен отладчик?**
   * Пошаговое выполнение кода
   * Просмотр значений переменных
   * Установка точек останова (**breakpoints**)
   * Анализ потока выполнения
2. **Ключевые функции:**
   * Запуск в режиме отладки
   * Шаг с заходом (**Step Into**) → F11
   * Шаг с обходом (**Step Over**) → F10
   * Продолжить выполнение (**Continue**) → F5

**Практическая часть**

**Задача 1: Отладка функции расчёта факториала**

**Код с ошибкой (Python):**

def factorial(n):

result = 1

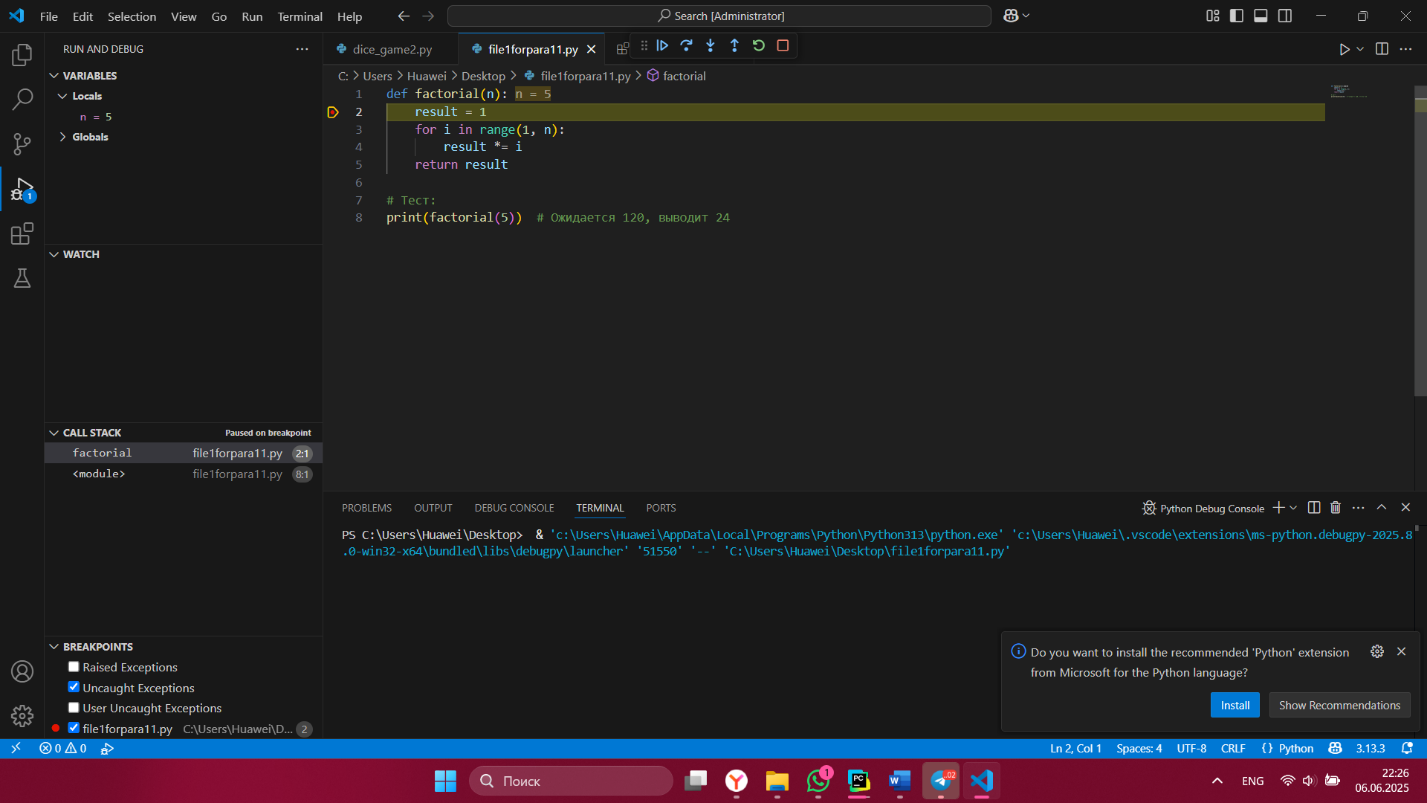
for i in range(1, n):

result \*= i

return result

*# Тест:*

print(factorial(5)) *# Ожидается 120, выводит 24*



Исправление ошибки

def factorial(n):

    result = 1

    for i in range(1, n + 1):  # Исправлено - добавлен +1

        result \*= i

    return result

print(factorial(5))  # Теперь выводит 120

**Задача 2: Работа с условиями**

**Код:**

def check\_password(password):

if len(password) < 8:

return "Слишком короткий!" *# Step Over*

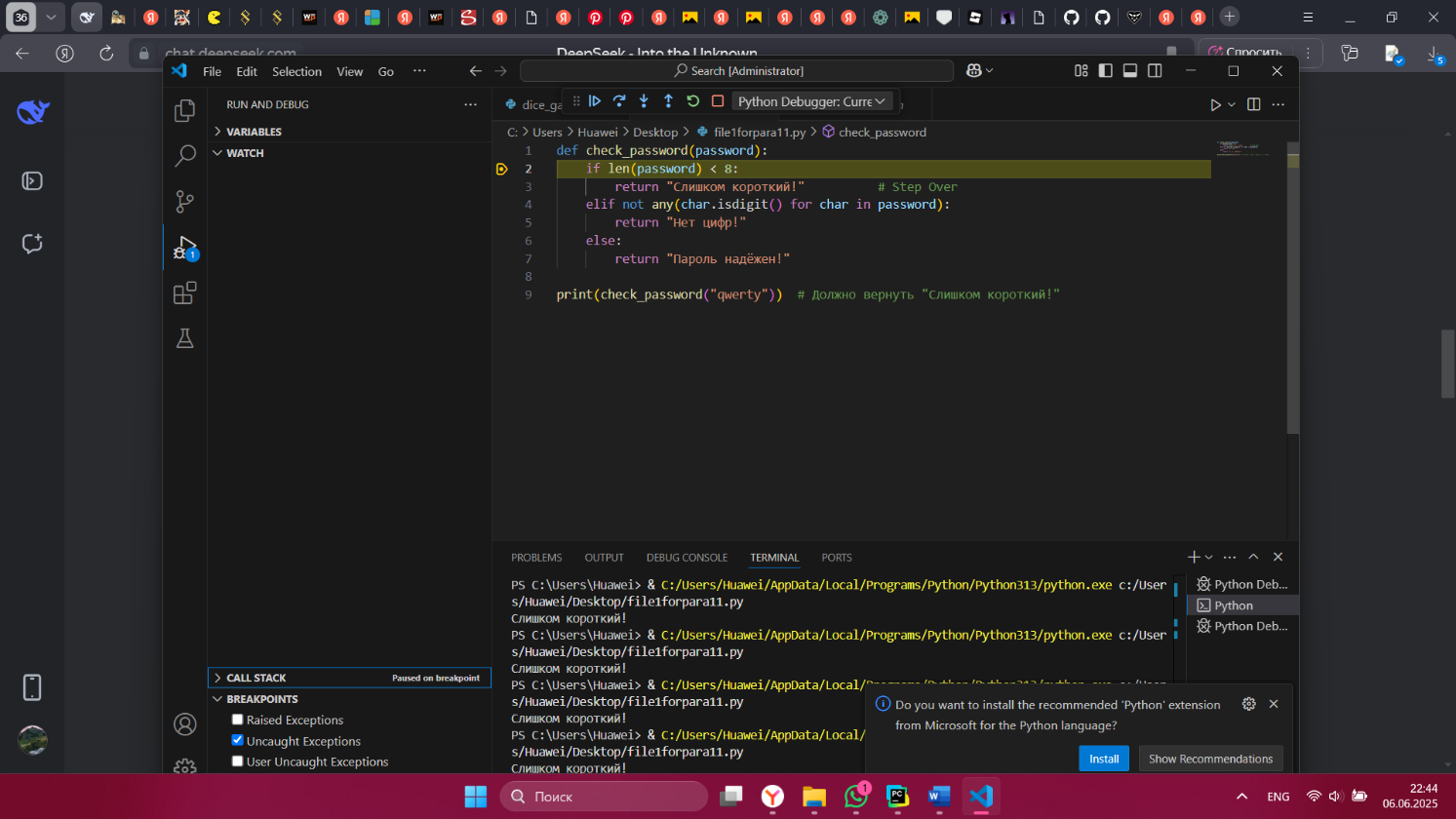
elif not any(char.isdigit() for char in password):

return "Нет цифр!"

else:

return "Пароль надёжен!"

print(check\_password("qwerty")) *# Должно вернуть "Слишком короткий!"*



**Задача 3: Работа с логикой**

**Код с ошибкой (Python):**

def calculate\_sum(arr):

total = 0

for i in range(0, len(arr) + 1): *# Ошибка: выход за границы массива*

total += arr[i]

return total

*# Тест:*

numbers = [10, 20, 30]

print(calculate\_sum(numbers)) *# Ожидается 60, но возникает IndexError*

Исправленный код

def calculate\_sum(arr):

total = 0

for i in range(0, len(arr)): # Исправлено: убран +1

total += arr[i]

return total

numbers = [10, 20, 30]

print(calculate\_sum(numbers)) # Теперь выводит 60

Ответы на вопросы отчёта:

1)Логические ошибки, Ошибки выхода за границы массивов, Ошибки в условиях циклов или ветвлений, Некорректные значения переменных.

2)Позволяют останавливать программу только при выполнении определённых условий.

Полезны для анализа конкретных сценариев и Экономят время, так как не требуют ручной проверки всех итераций.