Отчет по индивидуальному проекту: Применение отладчиков

Задача 1: Отладка функции расчёта факториала

Ошибочный код:

python

def factorial(n):

result = 1

for i in range(1, n):

result = i

return result

print(factorial(5)) Выводит 24, ожидается 120

Обнаружение ошибки:

- Установлена точка останова (breakpoint) на строке result = 1.

- Пошаговое выполнение через Step Over (F10).

- Выявлено, что цикл выполняется только до n - 1.

Исправленный код:

python

def factorial(n):

result = 1

for i in range(1, n + 1):

result = i

return result

print(factorial(5)) Теперь выводит 120

Задача 2: Работа с условиями

Код:

python

def check\_password(password):

if len(password) < 8:

return "Слишком короткий!"

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

return "Нет цифр!"

else:

return "Пароль надёжен!"

print(check\_password("qwerty")) Ожидается "Слишком короткий!"

Обнаружение и проверка:

- Установлена условная точка останова на строке if len(password) < 8: с условием: password == "qwerty".

- Отладчик останавливается только при передаче "qwerty" — это позволяет анализировать поведение для конкретного входного значения.

Результат:

- Функция корректно возвращает "Слишком короткий!", так как длина пароля меньше 8 символов.

Задача 3: Работа с логикой

Ошибочный код:

python

def calculate\_sum(arr):

total = 0

for i in range(0, len(arr) + 1): Ошибка здесь

total += arr[i]

return total

numbers = [10, 20, 30]

print(calculate\_sum(numbers)) IndexError

Обнаружение ошибки:

- Точка останова установлена на строке total = 0.

- Через Step Over (F10) отслеживались:

- значения i

- элементы массива arr[i]

- изменение переменной total

- Обнаружено: выход за границы массива — range() работает до len(arr) + 1.

Исправленный код:

python

def calculate\_sum(arr):

total = 0

for i in range(len(arr)): Исправлен диапазон

total += arr[i]

return total

numbers = [10, 20, 30]

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

Ответы на вопросы

1. Какие типы ошибок можно найти через отладчик?

- Логические ошибки — неправильная последовательность действий или условия.

- Ошибки выхода за границы массива — например, IndexError.

- Ошибка инициализации переменных — неверное начальное значение.

- Ошибки в циклах — неправильные условия завершения или диапазоны.

- Типизация и преобразование данных — несоответствие типов при операциях.

2. Чем полезны условные точки останова?

- Позволяют остановить выполнение программы только при определённом условии, например, при конкретном значении переменной.

- Это особенно удобно при работе с большими массивами, циклами или сложными состояниями, чтобы быстро находить проблемные участки кода без множества лишних шагов.

🖼️ Скриншоты отладчика:

Так как я не могу напрямую загрузить изображения, ниже вы можете добавить скриншоты из VS Code / PyCharm в ваш отчёт в Word. Вот рекомендации к ним:

- factorial\_debug.png: Отладчик на строке result = 1, показывает значения i и result до и после цикла.

- password\_condition\_breakpoint.png: Условная точка останова на if len(password) < 8: с фильтром password == "qwerty".

- calculate\_sum\_indexerror.png: Отладчик остановился на строке total += arr[i] с ошибкой IndexError.