Инструментальные средства разработки программного обеспечения.

Лабораторная работа: Отладка программы обработки данных на Python с использованием pdb.

Выполнено: Барсегян Дианой.

Цель работы: Изучение использования отладчика pdb для выявления и исправления ошибок в объектно-ориентированной программе на Python.

Задача: Программа Василия должна читать целые числа из файла data.txt, удваивать каждое значение и сохранять в файл processed_data.txt только значения, которые больше 10. Программа работает некорректно. Необходимо использовать pdb для выявления и исправления ошибки.

data.txt:

```
5
10
15
20
```

Был изменен вывод, в генерации списка к каждому элементу мы применяем метод strip() который отбрасывает все пустые значения в строке, а затем приводим это значение к int

```python
def load_data(filename):
    with open(filename, 'r') as file:
        data = file.readlines()
    return [int(x.strip()) for x in data]
```

Необходимо скорректировать функцию обработки данных:

C: > Users > Honor > Downloads > data_processor.py

```python
class DataProcessor:
    def __init__(self, data):
        self.data = data

    def process(self):
        self.data = [x * 2 for x in self.data]
        return self.data

    def filter(self, threshold):
        self.data = [x for x in self.data if x > threshold]
        return self.data
```

А так же функцию сохранения данных в новый файл:

C: > Users > Honor > Downloads > data_saver.py

```python
def save_data(filename, data):
    with open(filename, 'w') as file:
        for item in data:
            file.write(str(item) + '\n')

```

Шаги отладки с помощью pdb:

1. Запуск pdb: Точка останова добавлена в main() функцию.

2. Запуск скрипта: Запуск main.py останавливает выполнение в pdb.set_trace().

3. Использование команд pdb: Используем n (next), s (step), p (print), c (continue), l (list), q (quit).

Прохождение отладки и исправление ошибок:

• Ошибка 1: __init__: В DataProcessor был неправильно написан конструктор - init вместо __init__. Исправлено.

• Ошибка 2: Неправильное преобразование данных: В load_data данные не преобразовывались в целые числа, и символы новой строки не удалялись. Исправлено.

• Ошибка 3: Неправильная фильтрация: В методе filter не было условия фильтрации. Исправлено.

• Ошибка 4: Отсутствует символ новой строки: В методе save_data не добавлялся символ новой строки (\n), что приводило к слиянию чисел в одну строку. Исправлено.

Результат:

После исправления ошибок файл processed_data.txt содержит:

```
20
30
40
```

Вывод:

Использование pdb позволило эффективно идентифицировать и исправить несколько ошибок в программе, связанных с неправильным преобразованием данных, отсутствием фильтрации и неправильным использованием конструктора класса. Отладчик pdb является мощным инструментом для разработчиков Python.