**Куляхтина Алёна Алексеевна, группа БПИ204**

**Task 12 (Животные), function 10 (Straight Insertion)**

# **Отчет о написании программы**

**Общие сведения**

* Команды:
  + -i <input file path> - работа с данными из файла
  + -r <number of animals> - работа со сгенерированными данными

Примеры:

**-**i test/input3.txt

-r 15

* Обработка ошибок
  + При некорректном вводе параметров программа сообщает об этом и завершает работу.
  + При возникновении ошибок при чтении из файла программа выводит в консоль сообщение об ошибке с указанием номера текущей строки.
* Файлы с примерами вводимых данных находятся в папке tests и в названии содержат слово “input”. Файлы input2.txt и input3.txt направленны на проверку корректности работы сортировки контейнера. Файлы inputErr1.txt и inputErr2.txt показывают корректность работы обработки ошибок при чтении данных из файла.
* Запись отсортированных данных происходит в файл output.txt в папке test.

**Спецификации разработки**

* Операционная система: Ubuntu (64-bit)
* Архитектура: x86-64
* RAM: 15GB
* IDE: PyCharm (professional-2021.2.3)
* Примененные библиотеки
  + random
  + enum
  + time
  + string
  + argparse
  + abc
  + sys

**Структура проекта:**

* main.py – точка входа;
* animal.py – файл с описанием общего животного – родительского класса следующих;
* fish.py \ bird.py \ beast.py – файлы с описаниями рыб, птиц и зверей соответственно;
* container.py – структура контейнера для чтения и обработки животных;
* репозиторий venv – в этой папке содержатся файлы для работы виртуальной среды (сгенерированы автоматически при создании проекта)
* репозиторий tests:
  + input1.txt, …, input5.txt – файлы с тестовыми входными данными;
  + output.txt – файл для записи отсортированных данных;
  + inputErr1.txt, inputErr2.txt – файлы с некорректными данными.

**Характеристики проекта**

* Количество программных объектов – 6
* Размер используемых файлов - ~9.1Кб
* Размер проекта - ~ 10.6Мб
* Время генерации животных:

|  |  |
| --- | --- |
| * 10 элементов | * 0,01724 с |
| * 100 элементов | * 0,05518 с |
| * 1000 элементов | * 1,9384 с |
| * 10000 элементов | * 165,1088 с |

Вывод: программа, написанная на языке Python, работает медленнее, чем её аналоги, написанные ранее на языках C и C++. Стоит учесть, что данный язык программирования интерпретируемый, что в совокупности с динамической типизацией усложняет поиск более оптимальных способов решения.