**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе № 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Использование инструмента анализа качества | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Зайцев Никита Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Авдеенков Владимир Александрович |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г**

**Лабораторная работа №3.** **«Использование инструмента анализа качества»**

**Задания:**

* **Написать программу, генерирующую массив (минимум 10 элементов) вещественных чисел в диапазоне от –10 до 10 и определяющую минимальный положительный элемент;**

Листинг не эффективной программы b.py

import cProfile

from random import random, randint

def generate\_array(length):

array = []

for i in range(length+1):

val = round(randint(-10, 10) \* random(), 2)

array.append(val)

return array

def filter(array):

new\_array = []

for i in array:

if i > 0:

new\_array.append(i)

return new\_array

def search\_min\_pos(array):

min\_el = array[0]

for i in array:

if i < min\_el:

min\_el = i

return min\_el

def main():

length = 154\_349\_10

array = generate\_array(length)

filter\_array = filter(array)

min\_pos = search\_min\_pos(filter\_array)

print(f"Минимальное положительное число в массиве: {min\_pos}")

if \_\_name\_\_ == "\_\_main\_\_":

cProfile.run("main()")

**Листинг эффективной программы a.py**

import cProfile

from random import random, randint

def generate\_array(length: int) -> list[float]:

array = []

for \_ in range(length+1):

array.append(round(randint(-10, 10) \* random(), 2))

return array

def search\_min\_pos(array: list) -> float:

return min(filter(lambda x: x > 0, array))

def main():

length = 154\_349

array = generate\_array(length)

min\_pos = search\_min\_pos(array)

print(f"Минимальное положительное число в массиве: {min\_pos}")

if \_\_name\_\_ == "\_\_main\_\_":

cProfile.run("main()")

* Оценить эффективность разработанной программы (Т1);

Таб 1 — Таблица эффективности программ

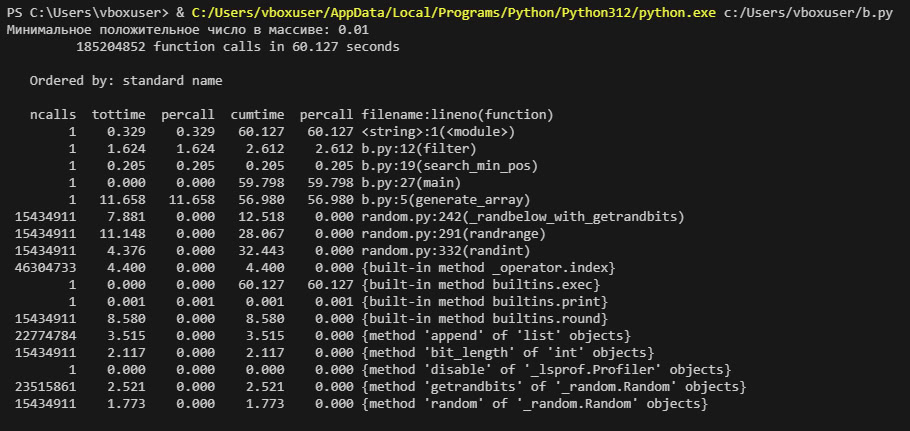
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Исходная программа(b.py) | | Улучшенная программа(a.py) | |
|  | Недостатки | Количественная оценка | Улучшения | Количественная оценка |
| Время выполнения | Использование циклов, вместо функций реализованных на Си(min) | 60.127 сек (рис 1) | Использование функции min | 55.973 сек (рис 2) |
| Оперативная память | Использование лишних структур | 29.3 Мб(рис 3) | По возможности использование генераторов списков | 29.0 Мб(рис 4) |
| Внешняя память | Использование множества циклов и функций  Отсутствие анонимных функций в местах, где они должны быть | 846Б(рис 5) | Сокращение до однострочников и использования внутренних функций  Появление lambda-функции для фильтрации | 637Б(рис 6) |

* Оценить качество разработанной программы (Т2);

Таб 3 — Показатели для b.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Правильность | Универсальность | Проверяемость | Точность результатов |
| Недостатки | Отсутствуют | Имеются ограничения по вводимому числу, а также отсутствует выбор шага. Выбор длины делается кодом | Отсутствуют | Отсутствуют |
| Оценка | Программа работает корректно по ТЗ | Все изменения производятся в коде, сами функции можно переиспользовать | Результат отличается о 0 на 0.1, что и является шагом программы | Точность гарантирован до десятых |

**Приложение**

Рис 1 — Время у b.py

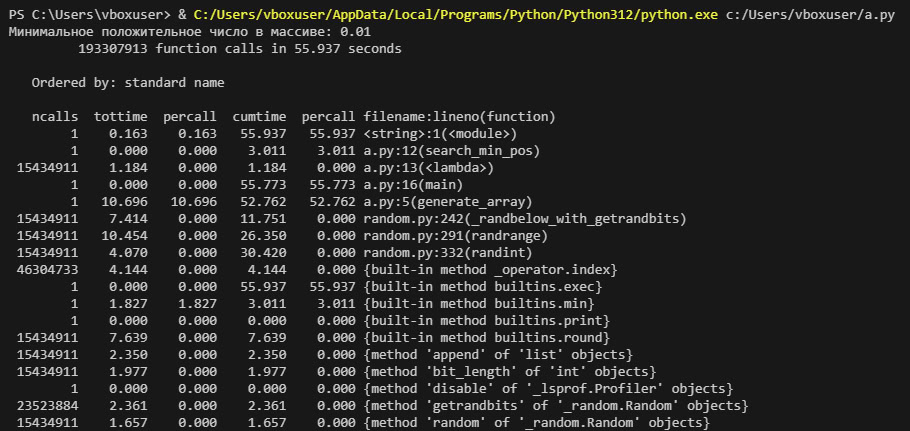
Рис 2 - Время у a.py

Рис 3 — Потребление ОЗУ у b.py

Рис 4 - Потребление ОЗУ у a.py

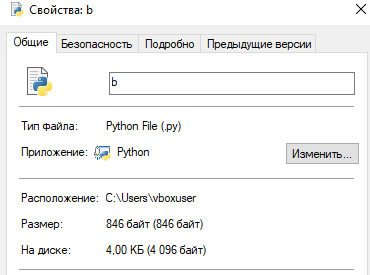


Рис 5 — Вес программы b.py

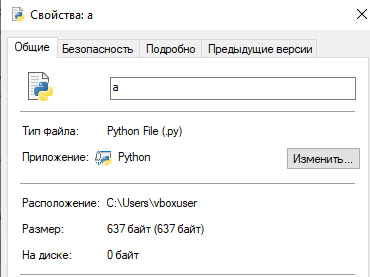


Рис 6 - Вес программы a.py