

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

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

Отчет по лабораторной работе № 3

| | |
|----------------|--|
| на тему | Использование инструмента анализа качества <small>(наименование темы)</small> |
|----------------|--|

| | |
|----------------------|--|
| по дисциплине | Тестирование информационных систем <small>(наименование дисциплины)</small> |
|----------------------|--|

| | | |
|--------------------|--|--------------------------|
| Обучающийся | Грачев Дмитрий Александрович <small>(ФИО)</small> | <small>(подпись)</small> |
|--------------------|--|--------------------------|

| | |
|---------------|----------|
| Группа | ДКИП-312 |
|---------------|----------|

| | | |
|----------------------|--|--------------------------|
| Преподаватель | Авдеенков Владимир Александрович <small>(ФИО)</small> | <small>(подпись)</small> |
|----------------------|--|--------------------------|

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 неэффективной программы o.py

```
1  from random import random, randint
2  from time import time
3  import os
4  from memory_profiler import memory_usage
5
6
7  def gen_arr(n):
8      arr = []
9      for _ in range(n+1):
10         arr.append(round(randint(-10, 10) * random(), 2))
11     min_num = 10 ** 20
12     for i in arr:
13         if i < min_num and i > 0:
14             min_num = i
15     return min_num
16
17 def measure_performance(length: int) -> tuple:
18     start = time()
19     obj = gen_arr(length)
20     end = time()
21     mem_usage = memory_usage((gen_arr, (length,)), max_usage=True)
22     execution_time = end - start
23     memory_used = mem_usage
24     return round(execution_time, 2), round(memory_used, 2)
25
26
27 if __name__ == "__main__":
28     length = 554_349_9
29     execution_time, memory_used = measure_performance(length)
30     print(f"Время выполнения: {execution_time} секунд")
31     print(f"Используемая память: {memory_used} МБ")
32     print(f"Файл занимает: {round(os.path.getsize("test_laba/random_array.py") / 1000, 2)} КБ")
```

Листинг эффективной программы random_array.py

```

1  from random import random, randint
2  from time import time
3  import os
4  from memory_profiler import memory_usage
5
6
7  class MinPositiveVal:
8      def __init__(self, length: int = 10) -> None:
9          self.length = length
10         self.array: list[float] = self.generate_array()
11
12
13     def generate_array(self) -> list[float]:
14         return [(round(randint(-10, 10) * random(), 2)) for _ in range(self.length+1)]
15
16
17     def search_min_pos(self) -> float:
18         return min(filter(lambda x: x > 0, self.array))
19
20
21     def __str__(self) -> str:
22         return str(self.search_min_pos())
23
24
25 def measure_performance(length: int) -> tuple:
26     start = time()
27     obj = MinPositiveVal(length)
28     end = time()
29     mem_usage = memory_usage((obj.search_min_pos, ), max_usage=True)
30     execution_time = end - start
31     memory_used = mem_usage
32     return round(execution_time, 2), round(memory_used, 2)
33
34
35 if __name__ == "__main__":
36     length = 554_349_9
37     execution_time, memory_used = measure_performance(length)
38     print(f"Время выполнения: {execution_time} секунд")
39     print(f"Используемая память: {memory_used} МБ")
40     print(f"Файл занимает: {round(os.path.getsize("test_laba/random_array.py") / 1000, 2)} КБ")

```

- Оценить эффективность разработанной программы (Т1);

Таб 1 — Таблица эффективности программ

| | Исходная программа(o.py) | | Улучшенная программа(random_array.py) | |
|------------------|--|-----------------------|--|-----------------------|
| | Недостатки | Количественная оценка | Улучшения | Количественная оценка |
| Время выполнения | Большое условие. Два цикла. Отсутствие встроенных функций | 5.48 сек (рис 1) | Использование функции min Использование ООП | 5.08 сек (рис 2) |

| | | | | |
|--------------------|--|------------------|--|------------------|
| | | | | |
| Оперативная память | Использование лишних структур | 235.01 Мб(рис 1) | По возможности использование генераторов списков | 234.75 Мб(рис 2) |
| Внешняя память | Отсутствие анонимных функций в местах, где они должны быть | 1.32КБ(рис 1) | Сокращение до однострочников и использования внутренних функций Появление lambda-функции для фильтрации | 1.32КБ(рис 2) |

- Оценить качество разработанной программы (Т2);

Таб 2 — Показатели

| | | | | |
|------------|------------------------------------|--|---|-----------------------------------|
| | 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 гарантирована до десятых |

Приложение

```
└─ /home/dimas/Projects/task_from_university/venv/bin/python /home/dimas/Projects/task_from_university/test_laba/o.py  
Время выполнения: 5.48 секунд  
Используемая память: 235.01 МБ  
Файл занимает: 1.32 КБ
```

Рис 1 — Показатели неэффективной программы o.py

```
└─ /home/dimas/Projects/task_from_university/venv/bin/python /home/dimas/Projects/task_from_university/test_laba/random_  
Время выполнения: 5.08 секунд  
Используемая память: 234.75 МБ  
Файл занимает: 1.32 КБ
```

Рис 2 — Показатели эффективной программы random_array.py