Итоговый отчет по лабораторной работе №1

**Название работы:** Ручное тестирование программного продукта (Вариант 9)  
**Студент:** Чернов В.В.  
**Дата:** 28.04.2025  
**Программа:** Управление памятью гибкого магнитного диска  
**Цель:** Освоение навыков ручного тестирования путем разработки и тестирования программы.

1. Описание программы

Программа, реализованная на Python, моделирует управление памятью гибкого магнитного диска объемом 360 КБ (368640 байт). Она поддерживает следующие функции:

* Запись файлов размером от 18 до 32768 байт в свободные блоки или в конец диска.
* Удаление файлов с освобождением памяти.
* Отображение статуса занятых и свободных блоков.
* Объединение смежных свободных блоков после удаления файлов.
* Интерактивное консольное меню для управления.

Программа использует динамические структуры данных (списки) для хранения занятых и свободных блоков памяти. Код доступен на GitHub.

2. План тестирования

План тестирования был разработан в соответствии с требованиями лабораторной работы и включал:

* **Цель:** Проверка соответствия программы требованиям варианта 9.
* **Функции для тестирования:** write\_file, delete\_file, get\_status, \_merge\_free\_blocks, main.
* **Методы:** Ручное функциональное тестирование, тестирование GUI (консольный вывод), тестирование граничных условий.
* **Документы:** План тестирования (TestPlan\_v1.docx), чек-лист (TestCases\_Checklist.docx), отчет об ошибках (BugReport.docx).
* **Сроки:** 13.04.2025 – 28.04.2025.

План доступен на GitHub.

3. Результаты тестирования

Тестирование проводилось согласно чек-листу, включающему 20 тестовых случаев (позитивные и негативные сценарии). Основные результаты:

* **Функциональность:** Все ключевые функции (write\_file, delete\_file, get\_status, \_merge\_free\_blocks) работают корректно:
  + Запись файлов выполняется в допустимом диапазоне размеров, с учетом фрагментации и нехватки места.
  + Удаление файлов освобождает память, с последующим объединением свободных блоков.
  + Статус диска отображается корректно.
  + Меню обрабатывает корректные опции и завершает работу при выборе выхода.
* **Обработка ошибок:** Программа корректно обрабатывает недопустимые размеры файлов, несуществующие файлы и нечисловой ввод для размера файла.
* **Граничные условия:** Проверены файлы минимального (18 байт) и максимального (32768 байт) размеров, а также переполнение диска.

Обнаруженные ошибки

| № | Функция | Описание | Серьезность | Статус |
| --- | --- | --- | --- | --- |
| 1 | write\_file | Отсутствие локализации сообщений об ошибках | Minor | Открыт |
| 2 | get\_status | Отсутствие локализации вывода статуса | Minor | Открыт |
| 3 | main | Аварийное завершение при пустом вводе в меню | Major | Открыт |

**Анализ:**

* Ошибки №1 и №2 (Minor) связаны с использованием английского языка вместо русского, что не влияет на функциональность, но может быть неудобно для русскоязычных пользователей.
* Ошибка №3 (Major) требует доработки, так как аварийное завершение программы при пустом вводе недопустимо. Рекомендуется добавить проверку на пустой ввод.

Полный отчет об ошибках доступен в файле BugReport.docx на GitHub.

4. Выводы

Программа успешно реализует требования варианта 9:

* Поддерживает управление памятью диска с использованием списков занятых и свободных блоков.
* Корректно обрабатывает запись, удаление файлов и отображение статуса.
* Обеспечивает объединение свободных блоков для оптимизации памяти.
* Имеет интерактивный интерфейс с обработкой большинства ошибок ввода.

**Оценка качества:**  
Функциональные требования выполнены полностью.  
Выявленные ошибки (2 Minor, 1 Major) не критичны для основной функциональности, но требуют доработки для повышения удобства использования и надежности.  
Тестирование проведено в полном объеме, все тестовые случаи задокументированы.

**Рекомендации:**

* Добавить локализацию сообщений на русский язык (ошибки №1, №2).
* Реализовать обработку пустого ввода в меню для предотвращения аварийного завершения (ошибка №3).
* Рассмотреть возможность добавления проверки на длину имени файла для предотвращения потенциальных проблем с длинными строками.

5. Структура

* **Код программы:** DiskMemoryManagement.py (https://github.com/DyavolZlozz/CHernovVV\_IVTz-211).
* **План тестирования:** TestPlan\_v1.docx.
* **Чек-лист:** TestCases\_Checklist.docx.
* **Отчет об ошибках:** BugReport.xlsx.
* **Итоговый отчет:** FinalReport.docx.

6. Заключение

Лабораторная работа выполнена в соответствии с требованиями. Программа протестирована, выявлены и задокументированы ошибки. Результаты тестирования подтверждают работоспособность программы и ее соответствие заданным требованиям.