**План тестирования программы управления памятью гибкого магнитного диска (Вариант 9)**

**1. Введение**  
Цель программы  
Программа моделирует управление памятью гибкого магнитного диска объемом 360 КБ (368640 байт). Она позволяет записывать файлы размером от 18 до 32768 байт, удалять их, отображать статус занятых и свободных участков памяти, а также объединять смежные свободные блоки для оптимизации пространства.  
Требования к программе (Use Case)

UC1: Пользователь может записать файл, указав имя и размер (18–32768 байт). Файл размещается в подходящем свободном блоке или в конце диска.  
UC2: Пользователь может удалить файл по имени, освобождая его место.  
UC3: Пользователь может запросить статус диска, отображающий занятые и свободные блоки.  
UC4: Программа корректно обрабатывает ошибки (недопустимый размер файла, отсутствие файла, нехватка места).  
UC5: Программа объединяет смежные свободные блоки после удаления файла.

**2. Содержание**  
2.1 Компоненты и функции для тестирования  
Программа реализована на Python и включает класс DiskManager и функцию main. Для тестирования выбраны следующие функции:

| # | Название приложения/компонента | Название функции | Ссылка на место в коде |
| --- | --- | --- | --- |
| 1 | DiskManager | write\_file(file\_name, file\_size) | DiskManager.write\_file |
| 2 | DiskManager | delete\_file(file\_name) | DiskManager.delete\_file |
| 3 | DiskManager | get\_status() | DiskManager.get\_status |
| 4 | DiskManager | \_merge\_free\_blocks() | DiskManager.\_merge\_free\_blocks |
| 5 | main | main() | main |

**3. Критерии качества**  
• Наличие плана тестирования с описанием тестовых случаев.  
• Чек-лист и тест-кейсы, покрывающие функциональные и нефункциональные требования.  
• Отчет об ошибках с указанием их серьезности.  
• Соответствие поведения программы заданным требованиям.

**4. Критические критерии успешности**  
• Корректная запись файлов в пределах допустимого размера и доступного пространства.  
• Корректное удаление файлов с освобождением памяти.  
• Правильное отображение статуса занятых и свободных блоков.  
• Объединение смежных свободных блоков после удаления файлов.  
• Обработка ошибок ввода (невалидный размер, несуществующий файл, переполнение диска).

**5. Ресурсы**  
5.1 Основные роли проекта

| # | Роль в проекте | Имя |
| --- | --- | --- |
| 1 | Менеджер проекта | Лавкина О.Ю. |
| 2 | Тестировщик | Чернов В.В. |

5.2 Оборудование и программное обеспечение

| # | Роль | Ресурс | Аппаратная конфигурация | Программная конфигурация |
| --- | --- | --- | --- | --- |
| 1 | Тестировщик | ПК | Intel Core i5, 8GB RAM | Windows 10, Python 3.9 |

5.3 Инструменты тестирования

| # | Инструмент | Описание |
| --- | --- | --- |
| 1 | Ручное тестирование | Взаимодействие с программой через консоль, проверка вывода и поведения функций |

**6. Документация по тестированию**

| # | Заголовок | Ответственное лицо | Периодичность | Способ публикации |
| --- | --- | --- | --- | --- |
| 1 | Отчет о тестировании | Чернов В.В. | Однократно | GitHub: |

**7. Стратегия тестирования**  
7.1 Критерии приостановки  
Тестирование приостанавливается в следующих случаях:  
• Обнаружение блокирующей ошибки, приводящей к аварийному завершению программы.  
• Критические сбои, такие как потеря данных или некорректное управление памятью, препятствующие дальнейшему тестированию.

7.2 Методы тестирования  
• Ручное функциональное тестирование: Проверка соответствия функций требованиям через ввод данных и анализ вывода.  
• Тестирование GUI: Проверка корректности отображения меню и сообщений в консоли.  
• Тестирование граничных значений: Проверка поведения на границах допустимых размеров файлов (18 байт, 32768 байт) и объема диска (368640 байт).

7.3 Типы тестов  
• Функциональное тестирование: Проверка функций write\_file, delete\_file, get\_status, \_merge\_free\_blocks и main.  
• Тестирование GUI: Проверка читаемости и корректности сообщений в консоли.  
• Тестирование граничных условий: Проверка поведения при минимальных, максимальных и недопустимых значениях.  
• Тестирование обработки ошибок: Проверка реакций на некорректный ввод (нечисловой размер, несуществующий файл).

7.4 Ошибки и документирование  
Ошибки классифицируются по серьезности:  
Blocker: Программа не запускается или аварийно завершается.  
Critical: Потеря данных, некорректное управление памятью.  
Major: Ошибки в размещении файлов или отображении статуса.  
Minor: Некритичные ошибки в сообщениях или форматировании.  
Trivial: Опечатки или косметические проблемы в выводе.  
Ошибки документируются в отчете (BugReport.docx) с указанием функции, описания, серьезности и статуса.

**8. План тестирования**

| # | Работы | Начало | Окончание | Задание | Расположение отчета |
| --- | --- | --- | --- | --- | --- |
| 1 | Разработка плана тестирования | 13.04.2025 | 13.04.2025 | TestPlan\_v1.docx | GitHub |
| 2 | Разработка чек-листа | 13.04.2025 | 17.04.2025 | TestCases\_Checklist.docx | GitHub |
| 3 | Тестирование и создание отчета об ошибках | 18.04.2025 | 28.04.2025 | BugReport.docx | GitHub |
| 4 | Повторное тестирование | 28.04.2025 | 28.04.2025 | FinalReport.docx | GitHub |

**9. Тестовые случаи (чек-лист)**  
Чек-лист включает позитивные и негативные тесты для всех функций. Примерный перечень:  
main: Проверка запуска, выбора опций меню, обработки некорректного ввода, выхода.  
write\_file: Запись файлов допустимых размеров (18, 32768 байт), недопустимых размеров (<18, >32768), при нехватке места, с нечисловым вводом.  
delete\_file: Удаление существующего и несуществующего файла.  
get\_status: Проверка статуса на пустом диске и с файлами.  
\_merge\_free\_blocks: Проверка объединения смежных и несмежных свободных блоков.  
Полный чек-лист доступен в файле TestCases\_Checklist.docx на GitHub.

**10. Примечания**  
• Тестирование проводится вручную через консольный интерфейс.  
• Программа не использует файловый ввод/вывод, все данные хранятся в оперативной памяти.  
• Результаты тестирования и артефакты будут опубликованы на GitHub:  
• В случае обнаружения ошибок они будут зафиксированы с указанием шагов воспроизведения и предложениями по исправлению.