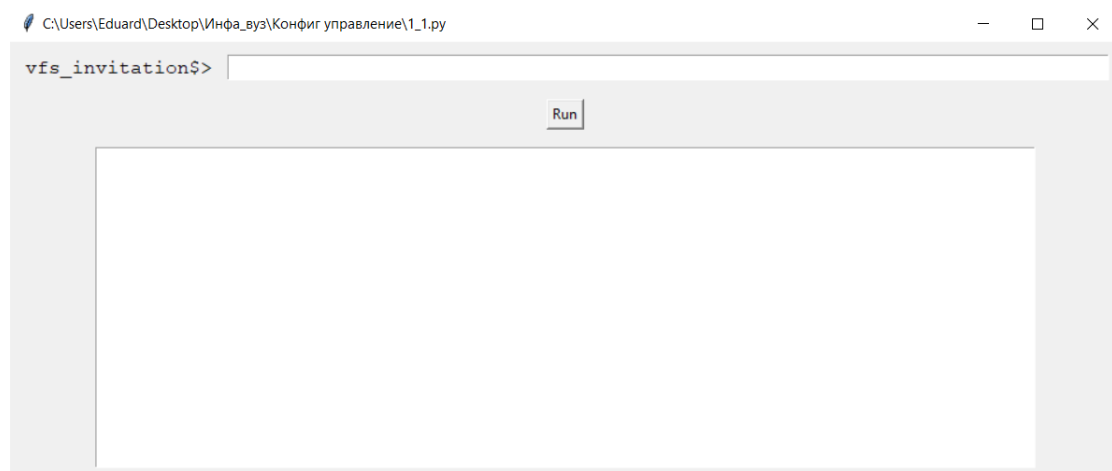


## Вариант 30

1 Этап: Мы выполнили данные задачи: Большинство функций в нем пока представляют собой заглушки, но диалог с пользователем уже поддерживается.

Требования:

1. Приложение должно быть реализовано в форме графического интерфейса (GUI).
2. Заголовок окна должен содержать имя VFS.
3. Реализовать простой парсер, который разделяет ввод на команду и аргументы по пробелам.
4. Сообщить об ошибке выполнения команд (неизвестная команда, неверные аргументы).
5. Реализовать команды-заглушки, которые выводят свое имя и аргументы: ls, cd.
6. Реализовать команду exit.
7. Продемонстрировать работу прототипа в интерактивном режиме. Необходимо показать примеры работы всей реализованной функциональности, включая обработку ошибок.



2 Этап: Цель: сделать эмулятор настраиваемым, то есть поддержать ввод параметров пользователя в приложение. Организовать для этого этапа отладочный вывод всех заданных параметров при запуске эмулятора.

Требования:

1. Параметры командной строки: – Путь к физическому расположению VFS. – Путь к стартовому скрипту.
2. Стартовый скрипт для выполнения команд эмулятора: поддерживает комментарии (используйте синтаксис из вашего языка реализации). При выполнении скрипта на экране отображается как ввод, так и вывод, имитируя диалог с пользователем.
3. Сообщить об ошибке во время исполнения стартового скрипта.
4. Создать несколько скриптов реальной ОС, в которой выполняется эмулятор. Включить в каждый скрипт вызовы эмулятора для тестирования всех поддерживаемых параметров командной строки.

```
1 @echo off
2 python 1_1.py test.txt
3 python 1_1.py
4 python 1_1.py test2.txt
5 pause
```