МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем

и технологий

**Отчет**

**по лабораторной работе № 4**

по дисциплине: ”Системное программирование”

на тему: ***”*Cкрипт-проект-диалоги*”***

Выполнил**:** студенты группы *10702121* Нестюк Е.А.

Огризко А.В.

Принял**:** пр. Давыденко Н. В.

Минск 2023

# Лабораторная работа №4.

**Цель работы:** Закрепить на практике принципы создания проектов с помощью скриптов Shell, освоить средства примитивного графического интерфейса в сприктах Shell.

## Задание 1

1. Cделать пример с двумя вложенными диалогами типа YesNo.

2. Cделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя+профессию, прочитанную в диалоге.

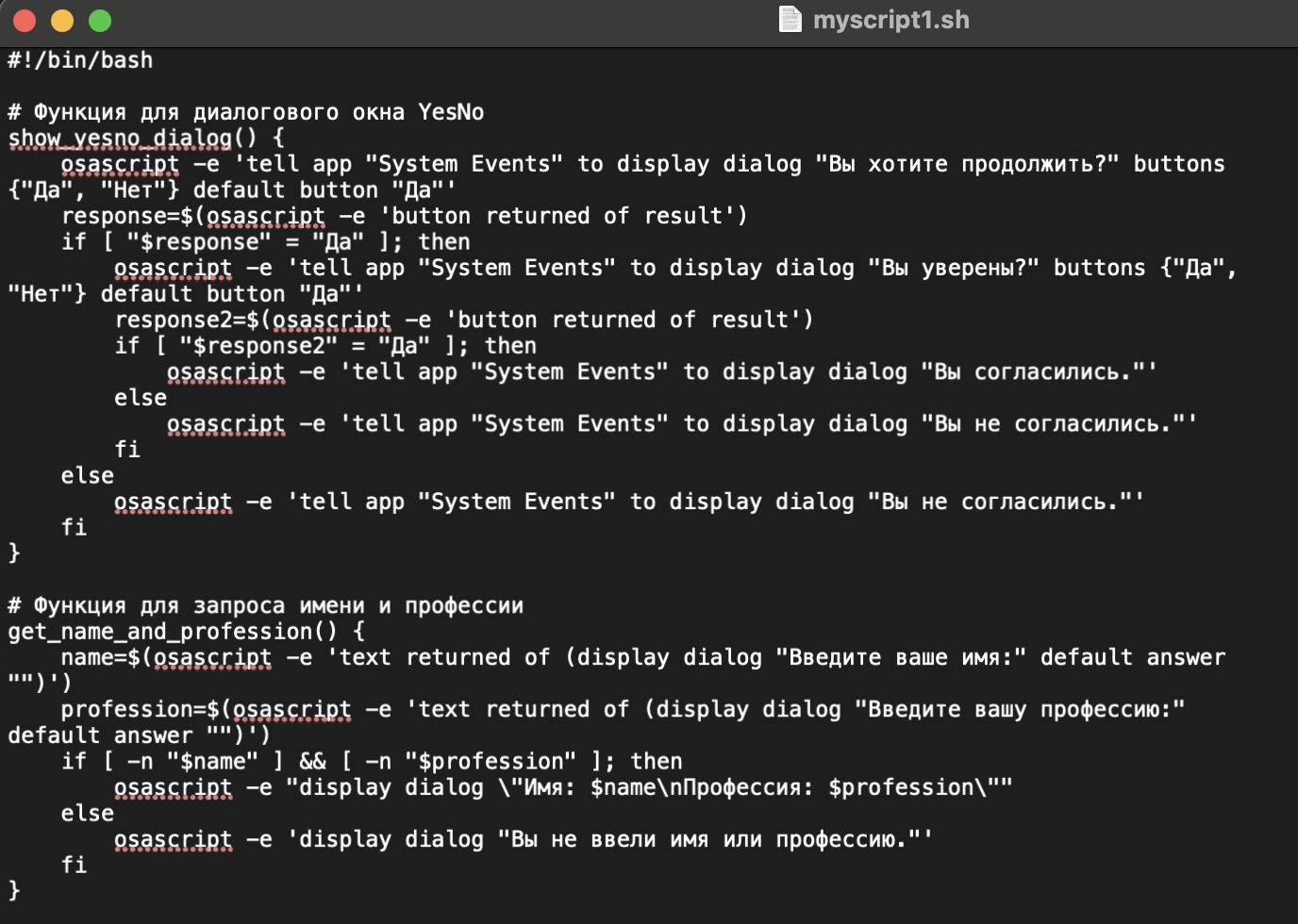
3. Вывести список с названиями валют. После выбора валюты система должна вывести ее котировку.

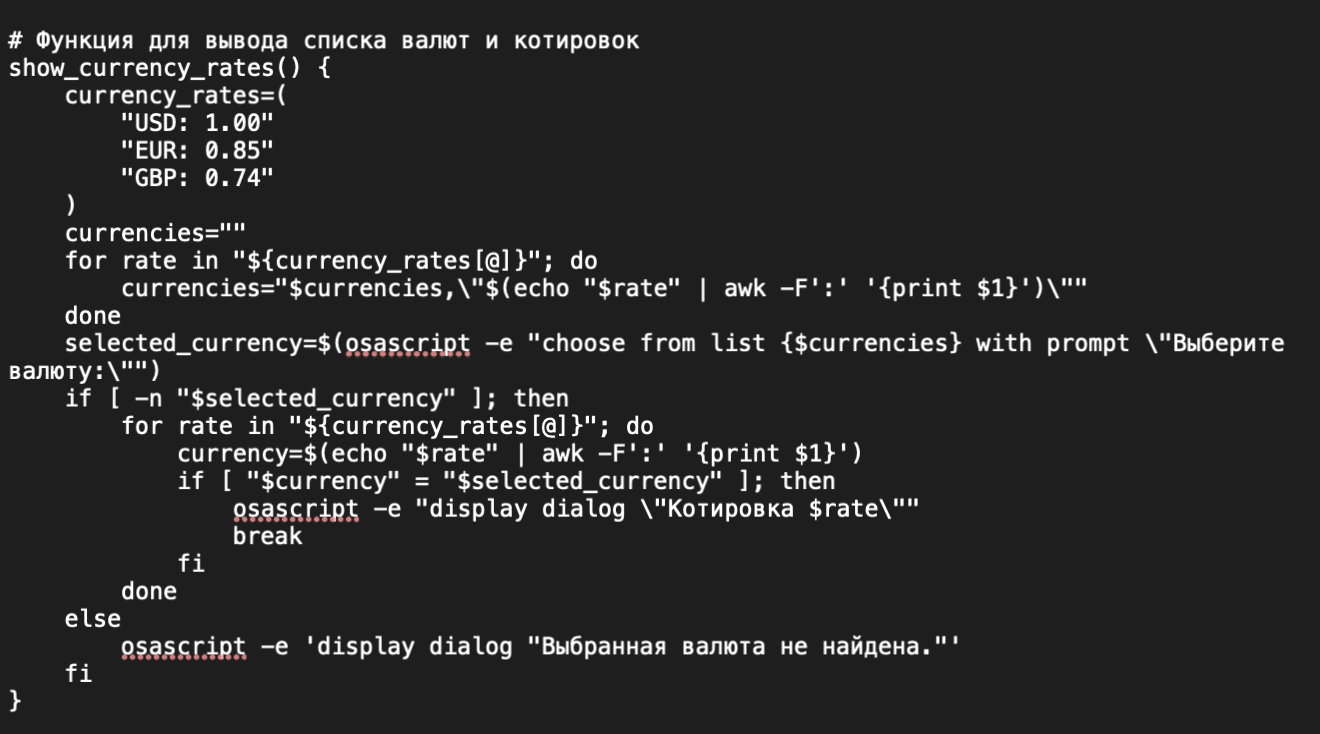
4. Сделать оконное меню валют к 3 пункту.

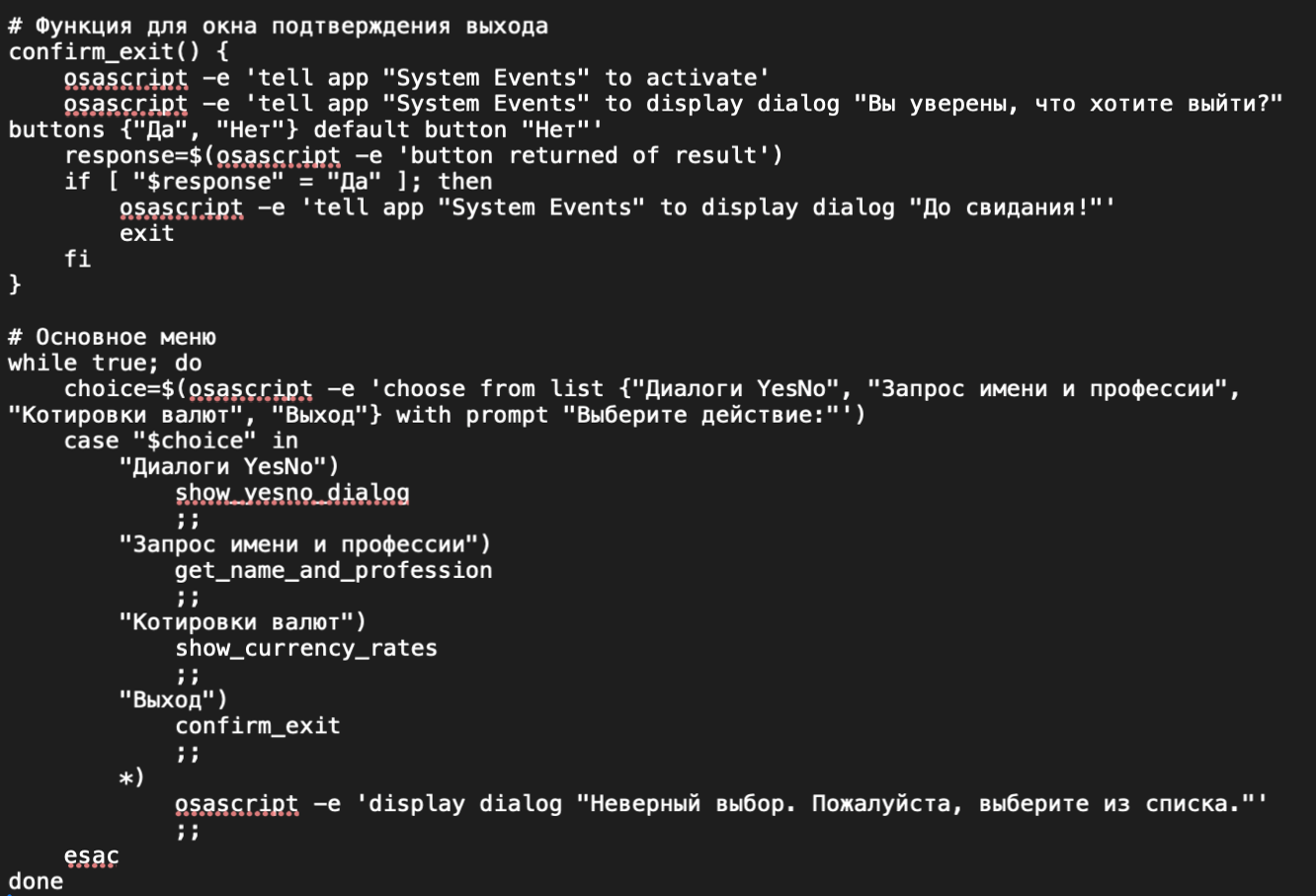
5. Добавить выход из программы.

**Решение**

*1.Пишем скрипт с двумя вложенными диалогами типа YesNo сразу вместе с элементами для пункта 2, 3, 4, 5.*

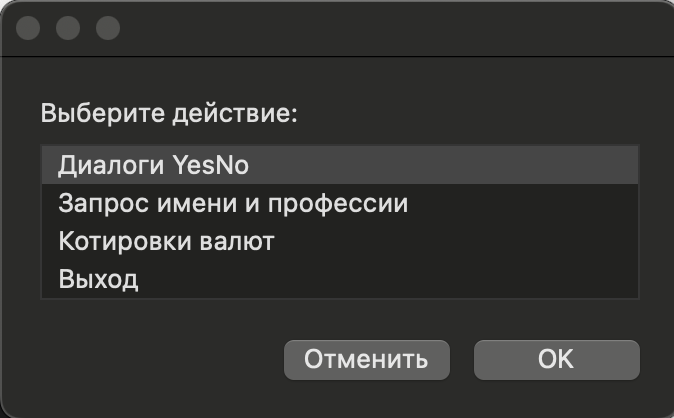




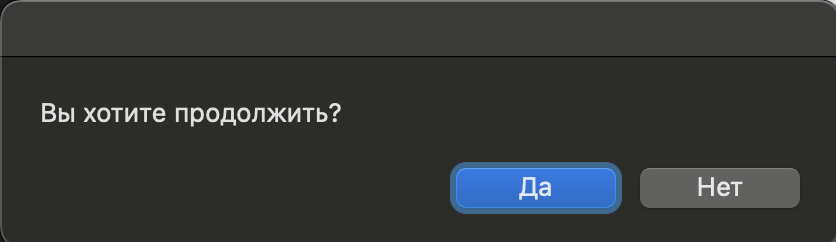




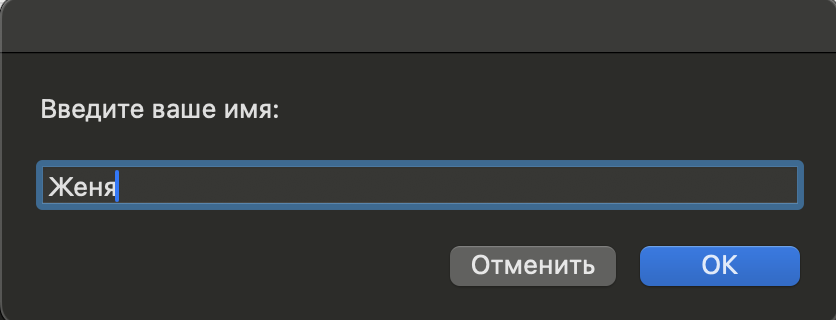
*2.Основное меню*

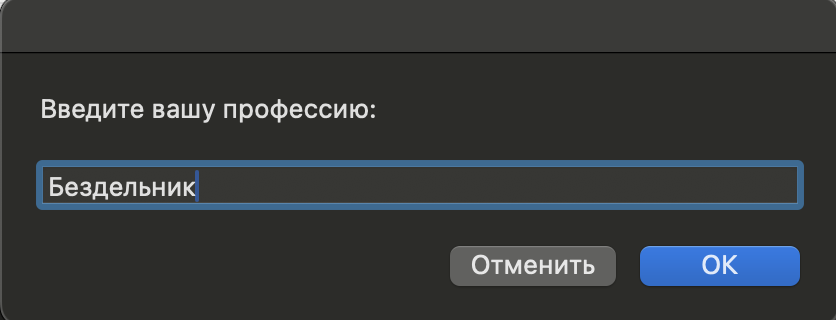
****

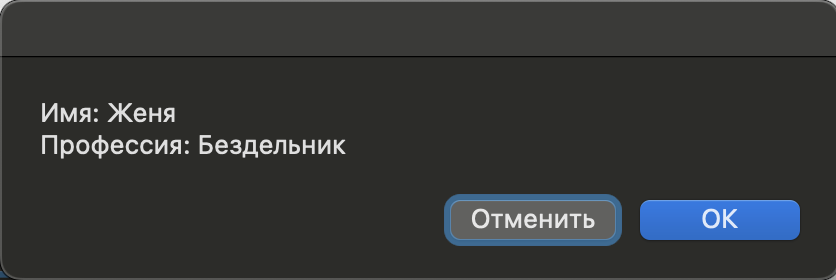
*3.Диолговое окно YesNo*



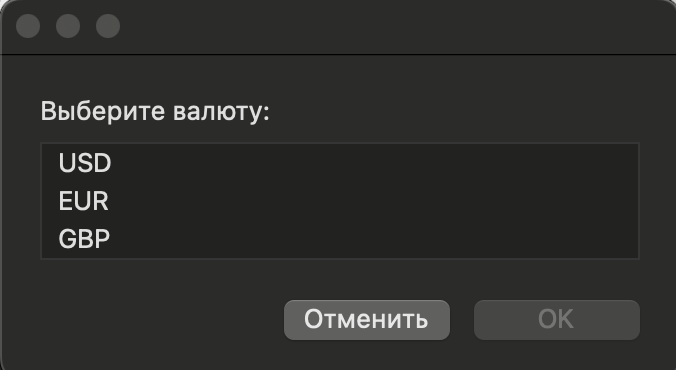
*4.Запрашиваем на ввод имя человека и профессии*

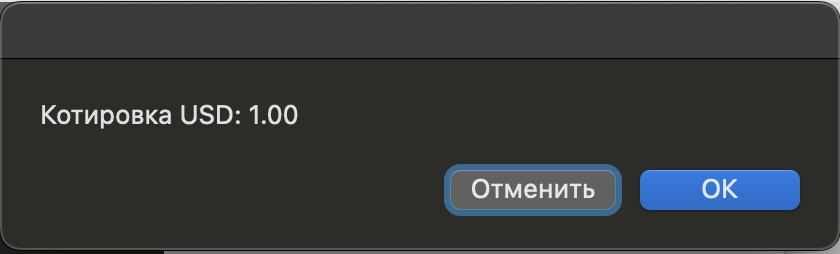


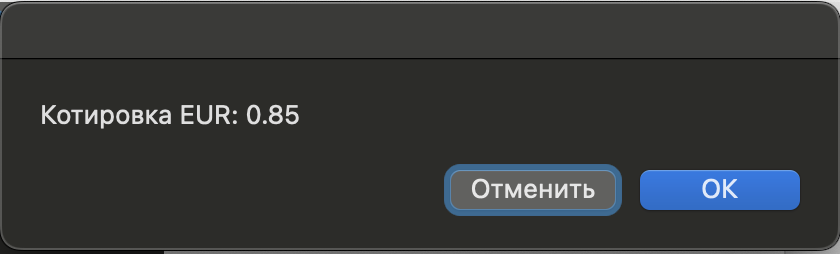




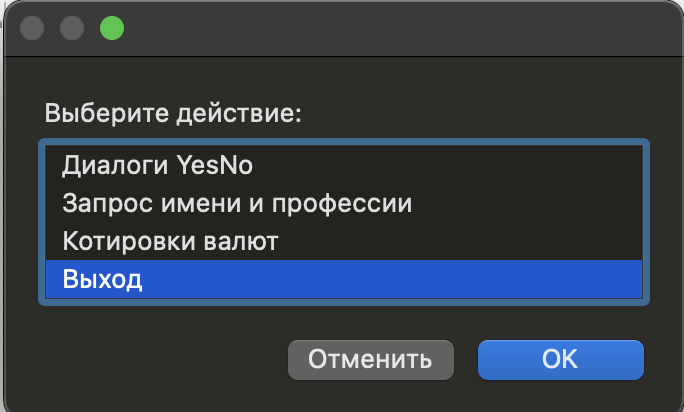
*5. Список с названиями валют. Выбора валюты с выводом её котировки. Оконное меню валют.*

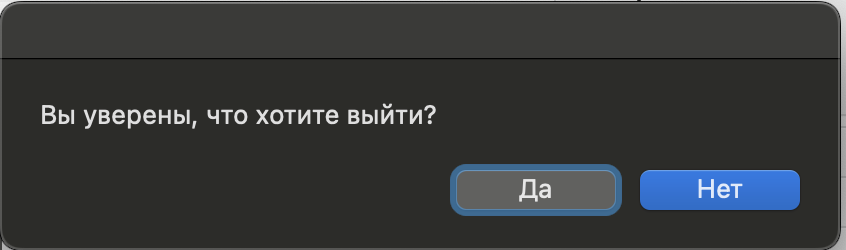






6.Выход из программы с окном подтверждения

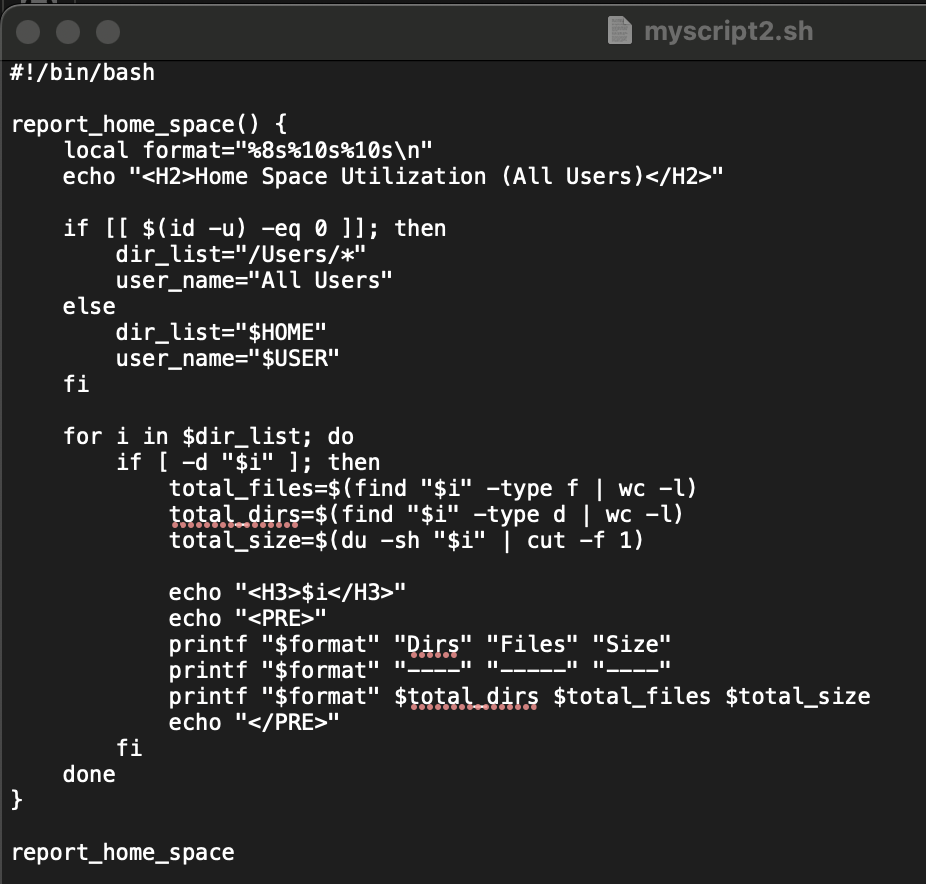




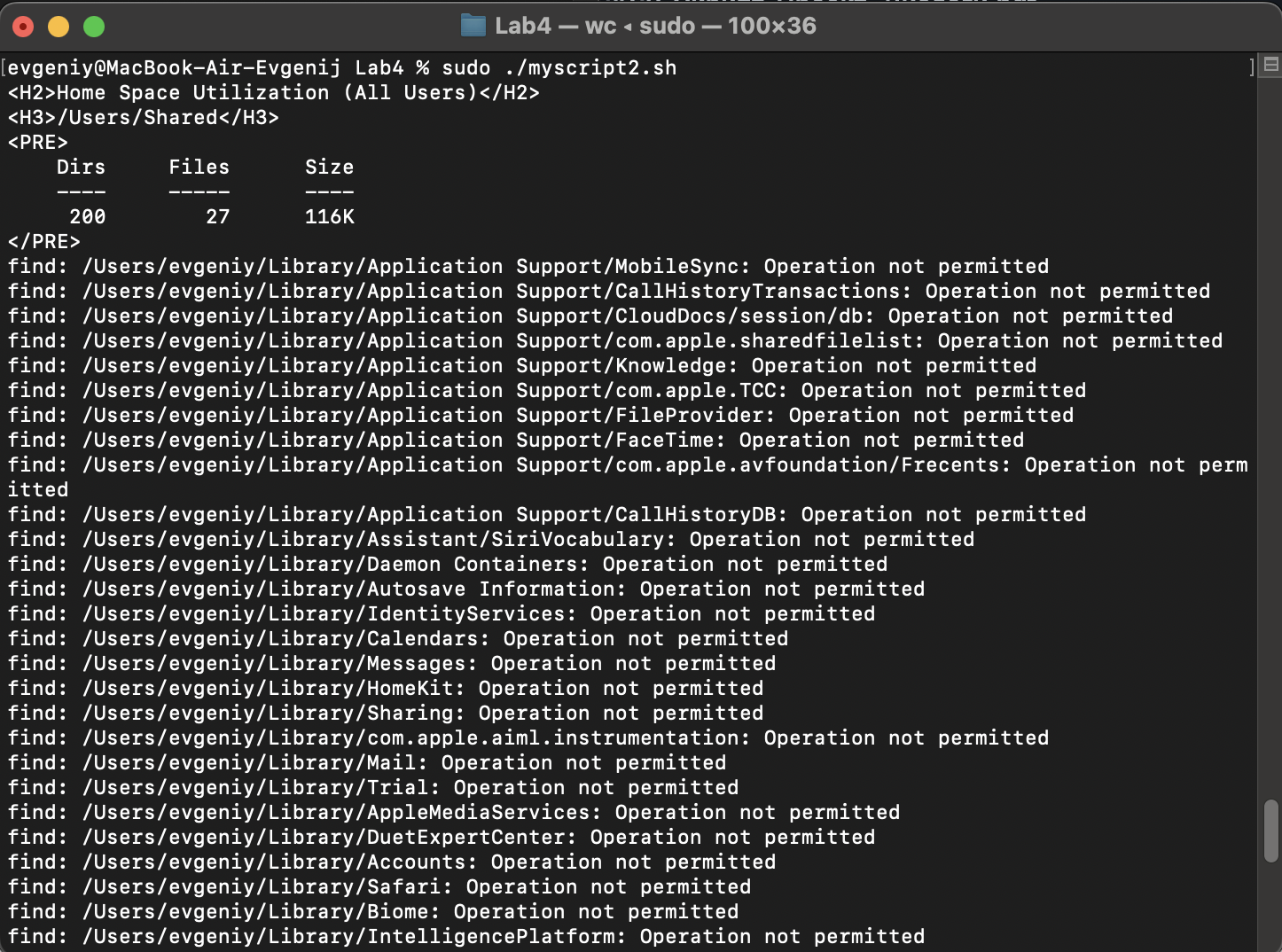
**Задание 2.**

Создайте копию проекта представленного в упражнении №3. Внесем изменения в новом проекте, добавив вывод информации о домашнем каталоге каждого пользователя и включив в вывод общее число файлов и подкаталогов в каждом из них.

*1. Вносим изменения в новом проекте*



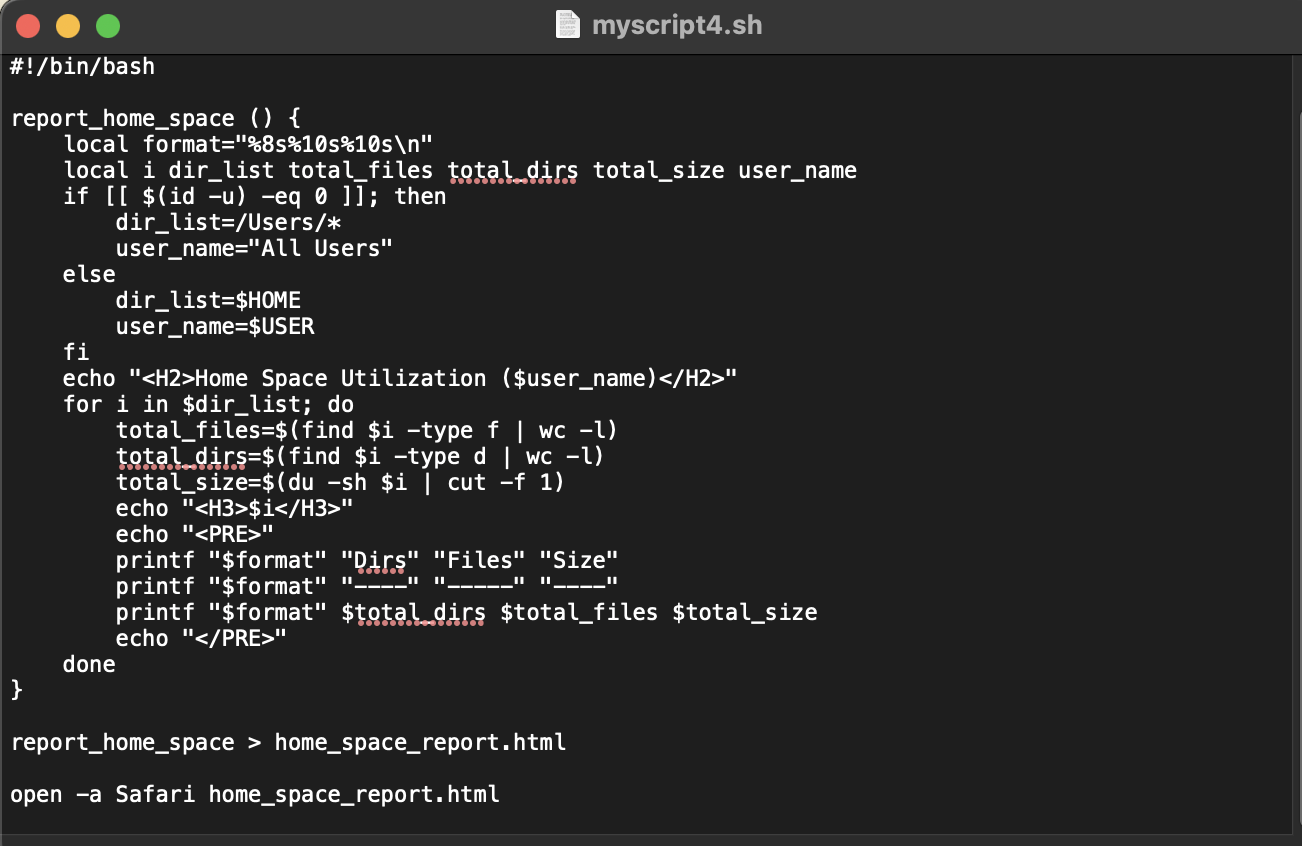
*5. Вывод информации о домашних каталогах пользователей и общем количестве файлов и подкаталогов в каждом из них.*

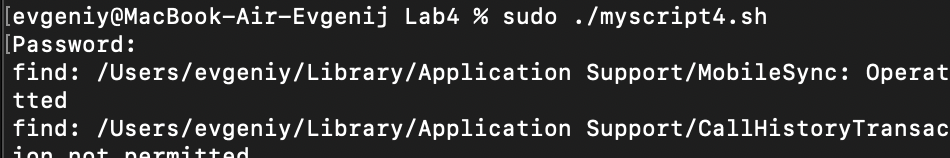


**Задание 3.**

Cоздайте графическое диалоговое окно. Информация должна автоматически передаваться в HTML-файл и открываться в браузере.

*1. Cоздание графического диалогового окна с запись в HTML-файл и открытием в браузере*

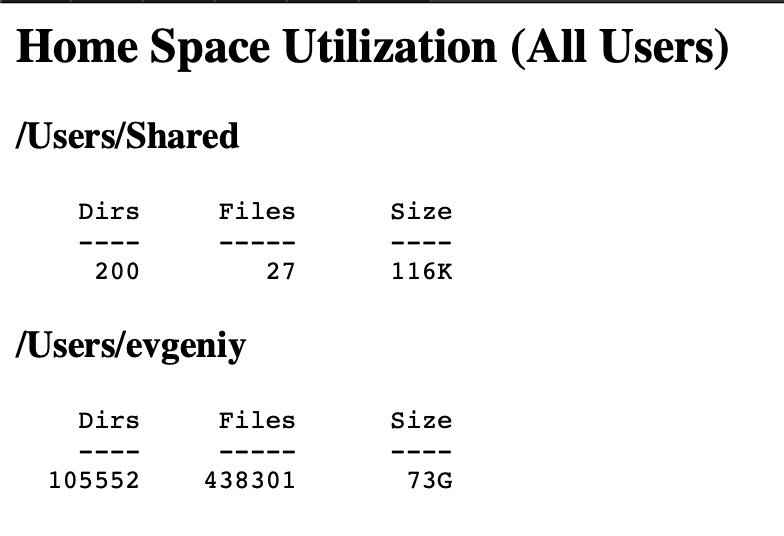




*2. Cоздание графического диалогового окна с запись в HTML-файл и открытием в браузере*



*3. Наблюдаем за итоговым результатом в браузере*



**Контрольные вопросы:**

**Возможности командной оболочки и скриптов Shell**

3) Какое типичное соглашение о написании имен переменных и констрант принято программистами? Для чего это делается?

Программисты, работая с командной оболочкой и написанием скриптов на языке командной оболочки (например, Bash), обычно придерживаются следующих соглашений относительно имен переменных и констант:

1. Верхний регистр для констант: Имена констант (переменных, значения которых не изменяются) обычно записываются заглавными буквами с подчеркиваниями в качестве разделителей. Пример: MY\_CONSTANT.
2. Нижний регистр для переменных: Имена переменных обычно записываются в нижнем регистре с подчеркиваниями в качестве разделителей. Пример: my\_variable.
3. CamelCase для переменных: Вместо подчеркиваний, некоторые программисты предпочитают использовать CamelCase для имен переменных. Пример: myVariable.
4. Одиночное подчеркивание для частных переменных: Иногда используется одиночное подчеркивание в начале имени переменной для обозначения приватных переменных, которые не должны напрямую изменяться извне. Пример: \_private\_variable.
5. Двойное подчеркивание для магических методов: В Python, для именования магических методов используется двойное подчеркивание. Пример: \_\_init\_\_.

Соблюдение этих соглашений помогает делать код более читаемым и понятным, как для вас, так и для других разработчиков. Это также уменьшает вероятность конфликтов и путаницы между именами переменных и констант.

Соглашения об именах переменных и констант - это часть хороших практик программирования и способ сделать ваш код более поддерживаемым и легко читаемым для других разработчиков.

**Графический интерфейс диалога**

13) Какие существуют типы окон примитивного графического интерфейса Linux-скриптов?

В Linux-скриптах существует несколько типов окон примитивного графического интерфейса, которые могут быть реализованы с помощью утилит, таких как Zenity, Yad и Dialog. Некоторые из распространенных типов окон включают в себя:

1. Диалоговое окно с информацией (Info Dialog): Отображает информационное сообщение или уведомление без кнопок или возможности взаимодействия с пользователем.

Пример: zenity --info --text="Привет, мир!"

1. Диалоговое окно подтверждения (Confirmation Dialog): Запрашивает у пользователя подтверждение или выбор между двумя вариантами (например, "Да" или "Нет").

Пример: zenity --question --text="Вы согласны?"

1. Диалоговое окно с вводом данных (Entry Dialog): Запрашивает у пользователя ввод данных (текста) с возможностью сохранения этого введенного значения.

Пример: zenity --entry --text="Введите ваше имя:"

1. Диалоговое окно выбора файла или директории (File/Directory Selection Dialog): Позволяет пользователю выбрать файл или директорию на компьютере.

Пример: zenity --file-selection --title="Выбор файла"

1. Диалоговое окно списка или меню выбора (List/Menu Selection Dialog): Предоставляет пользователю список вариантов для выбора.

Пример: zenity --list --title="Выберите цвет" --column="Имя" --column="Цвет" Ян "Красный" Лена "Синий"

Это лишь некоторые из типов окон, доступных для использования в скриптах с помощью утилит примитивного графического интерфейса в Linux.