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: ”Скрипты, создание проекта, диалоги ***”***

Выполнил**:** студент группы 10701321 Нестерков Д.А.

Принял**:** пр. Давыденко Н.

Минск 2023

# **Лабораторная работа № 4**

**Цель работы**

Закрепить на практике принципы создания проектов с помощью скриптов SHELL, освоить средства примитивного графического интерфейса в Linux-скриптах.

**Задание**

**Задание 1**

1. Сделать пример с двумя вложенными диалогами типа YesNo.

2. Сделать пример, запрашивающий сначала имя человека, а потом профессию. Вывести имя + профессию, прочитанные в диалоге.

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

Здесь следует обратить внимание на проверку условия. Запомните, если выполняем сравнение строк, то в квадратных скобках строки записываем в двоичных кавычках. При этом (!) нужно разделять все слова пробелами как слева, так и справа. Наконец, слово then договоримся писать на новой строке. Заметим, что диалог отображается списком, поскольку указаны символы перехода на новую строку в команде dialog.

4. Измените предыдущую программу так, вместо списка валют предлагалось оконное меню валют. Чтобы программа работала в цикле. Для выхода из цикла нужно вместо названия валюты вводить exit.

5. Измените предыдущую программу так, для выхода из программы в меню была кнопка закрытия программы «exit».

Работа диалогов должна быть подтверждена скриншотами в отчетах.

**Задание 2**

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

В этой новой версии проверяется наличие привилегий суперпользователя, но вместо того, чтобы выполнить полный набор операций в каждой из ветвей if, здесь устанавливаются некоторые переменные, которые затем используются в цикле for. В функции использованы несколько локальных переменных и команда printf для форматирования части вывода.

**Задание 3**

Для сценария выполненного в задании №2 создайте графическое диалоговое окно.

**Результаты выполнения**

**Задание 1**

#!/bin/bash

# Устанавливаем XQuartz как оконную систему по умолчанию

export DISPLAY=:0

# Запрашиваем первый ответ

answer1=$(zenity --entry --text="Да или нет?" --title="Диалоговое окно" --width=300)

# Проверяем ответ

if [ "$answer1" == "да" ]; then

# Запрашиваем второй ответ

answer2=$(zenity --entry --text="Вы уверены?" --title="Диалоговое окно" --width=300)

# Проверяем второй ответ

if [ "$answer2" == "да" ]; then

# Показываем результат

zenity --info --text="Вы согласились дважды." --title="Результат" --width=300

else

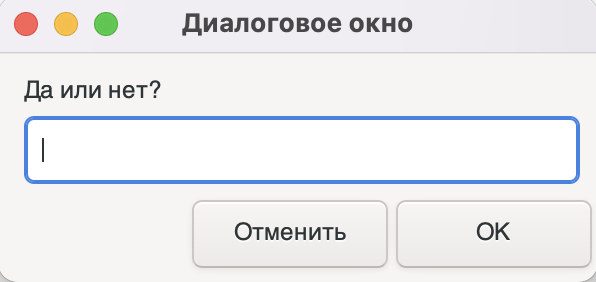
zenity --info --text="Вы изменили решение." --title="Результат" --width=300

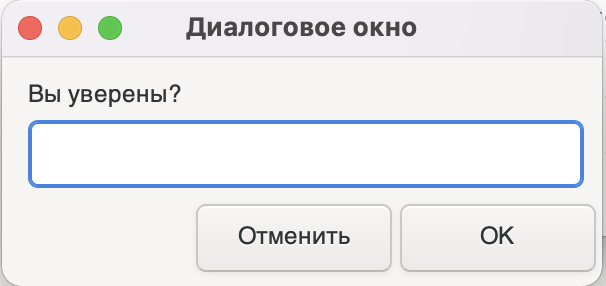
fi

else

zenity --info --text="Вы отказались." --title="Результат" --width=300

fi





Рисуноки 1-2 Два диалоговых окна

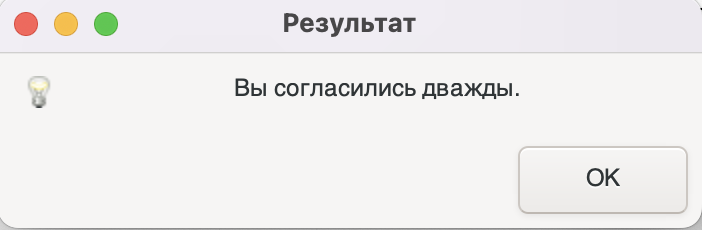


Рисунок 3 Результат

#!/bin/bash

# Устанавливаем дисплей для zenity

export DISPLAY=:0

# Запрашиваем имя

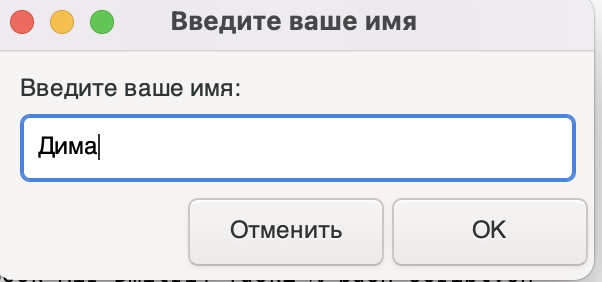
name=$(zenity --entry --title="Введите ваше имя" --text="Введите ваше имя:" --width=300)

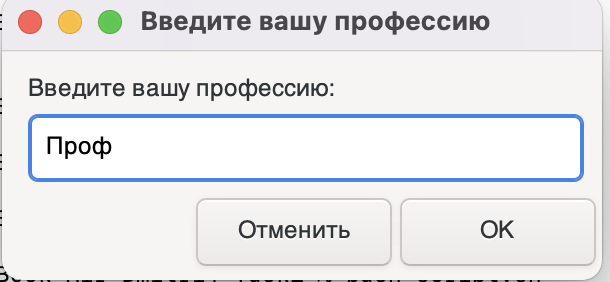
# Запрашиваем профессию

profession=$(zenity --entry --title="Введите вашу профессию" --text="Введите вашу профессию:" --width=300)

# Показываем результат

zenity --info --title="Результат" --width=300 --text="Имя: $name\nПрофессия: $profession"





Рисунки 3-4. Скрипт для запроса имени и профессии

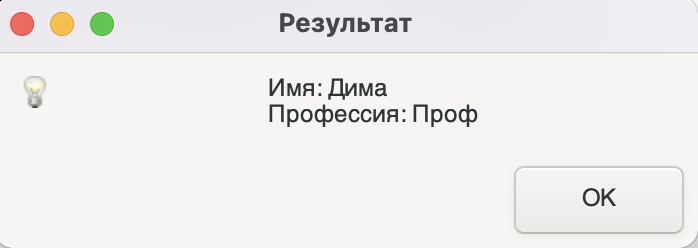


Рисунок 5 Результат

**#!/bin/bash**

**# Экспортируем дисплей для zenity**

**export DISPLAY=:0**

**# Массив доступных валют**

**currencies=("USD" "EUR" "GBP" "Выход")**

**# Показываем список через zenity**

**choice=$(zenity --list --title="Выберите валюту" --text="Валюта:" --column="Валюта" "${currencies[@]}" --width=300)**

**# Проверяем выбранный пункт**

**case $choice in**

**"USD")**

**zenity --info --title="Котировка USD" --text="Котировка USD: 3.3" --width=300**

**;;**

**"EUR")**

**zenity --info --title="Котировка EUR" --text="Котировка EUR: 3.4" --width=300**

**;;**

**"GBP")**

**zenity --info --title="Котировка GBP" --text="Котировка GBP: 3.6" --width=300**

**;;**

**"Выход")**

**exit 0**

**;;**

**\*)**

**zenity --error --title="Ошибка" --text="Выбранной валюты нет в списке." --width=300**

**;;**

**Esac**

**С циклом**

**#!/bin/bash**

**# Экспортируем дисплей для zenity**

**export DISPLAY=:0**

**while true; do**

**# Показываем список валют**

**choice=$(zenity --list --column="Валюта" "USD" "EUR" "GBP" "Выход" --width=300)**

**case $choice in**

**"Выход")**

**exit 0**

**;;**

**"USD")**

**zenity --info --text="Котировка USD: 3.3" --width=300**

**;;**

**"EUR")**

**zenity --info --text="Котировка EUR: 3.4" --width=300**

**;;**

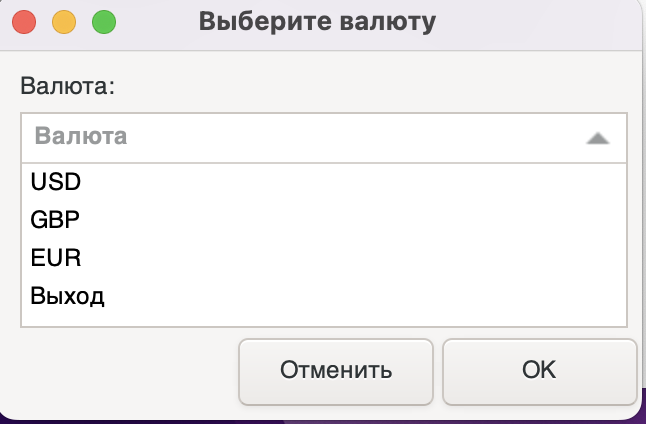
**"GBP")**

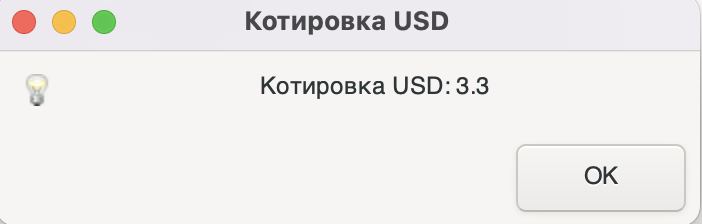
**zenity --info --text="Котировка GBP: 3.6" --width=300**

**;;**

**esac**

**Done**





Рисунки 6-7. Скрипт для вывода списка валют и их котировок

**Задание 2**

**#!/bin/bash**

**report\_home\_space () {**

**local format="%-30s%-10s%-10s%-10s\n"**

**local dir\_list total\_files total\_dirs total\_size**

**# Для macOS используем другой путь к домашним каталогам**

**dir\_list="/Users/\*"**

**echo "<H2>Home Space Utilization</H2>"**

**echo "<PRE>"**

**printf "$format" "User" "Dirs" "Files" "Size"**

**printf "$format" "----" "----" "-----" "----"**

**for i in $dir\_list; do**

**total\_files=$(find "$i" -type f | wc -l)**

**total\_dirs=$(find "$i" -type d | wc -l)**

**total\_size=$(du -sh "$i" | cut -f 1)**

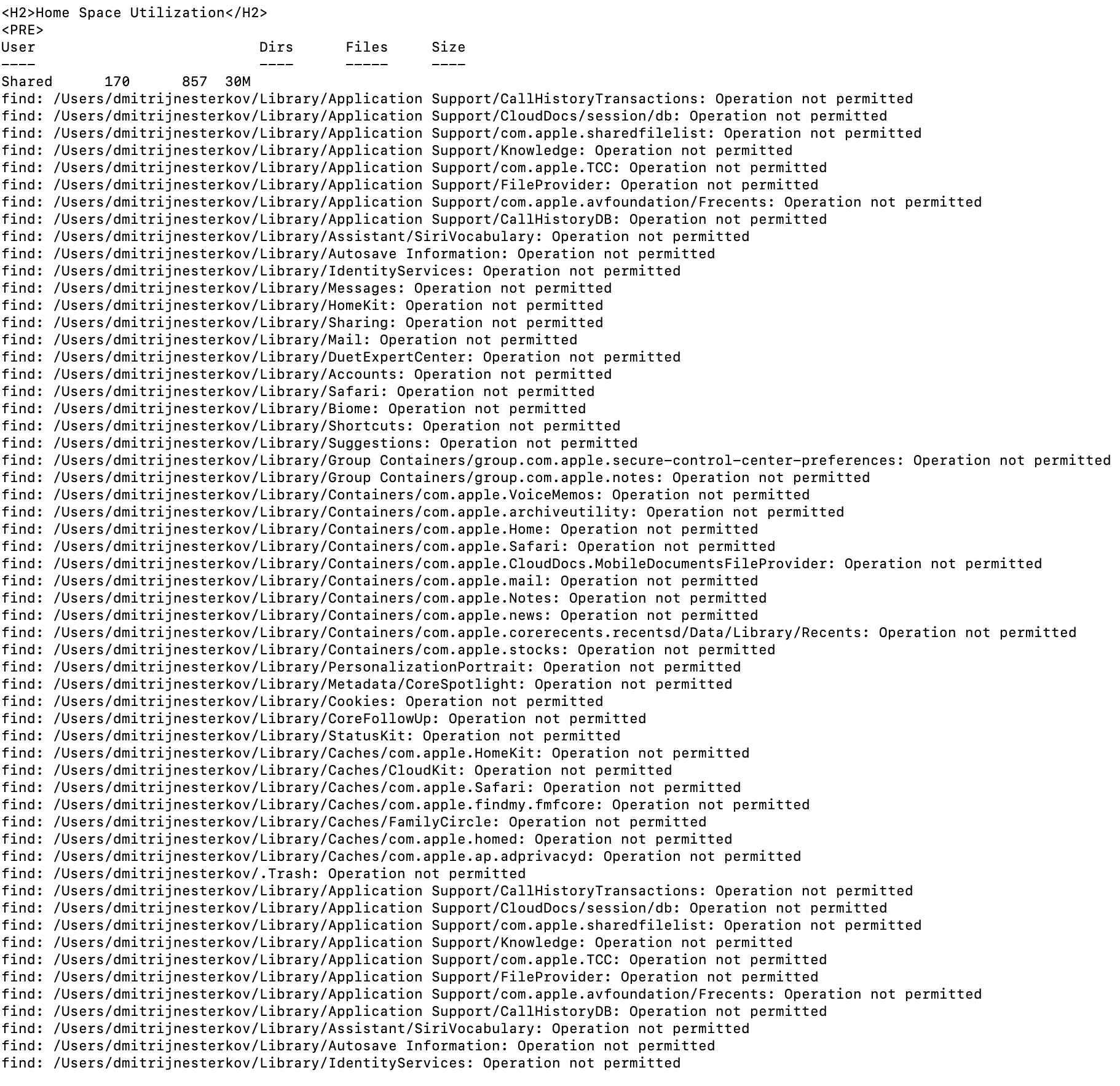
**echo "$(basename $i) $total\_dirs $total\_files $total\_size"**

**done**

**echo "</PRE>"**

**}**

**report\_home\_space**



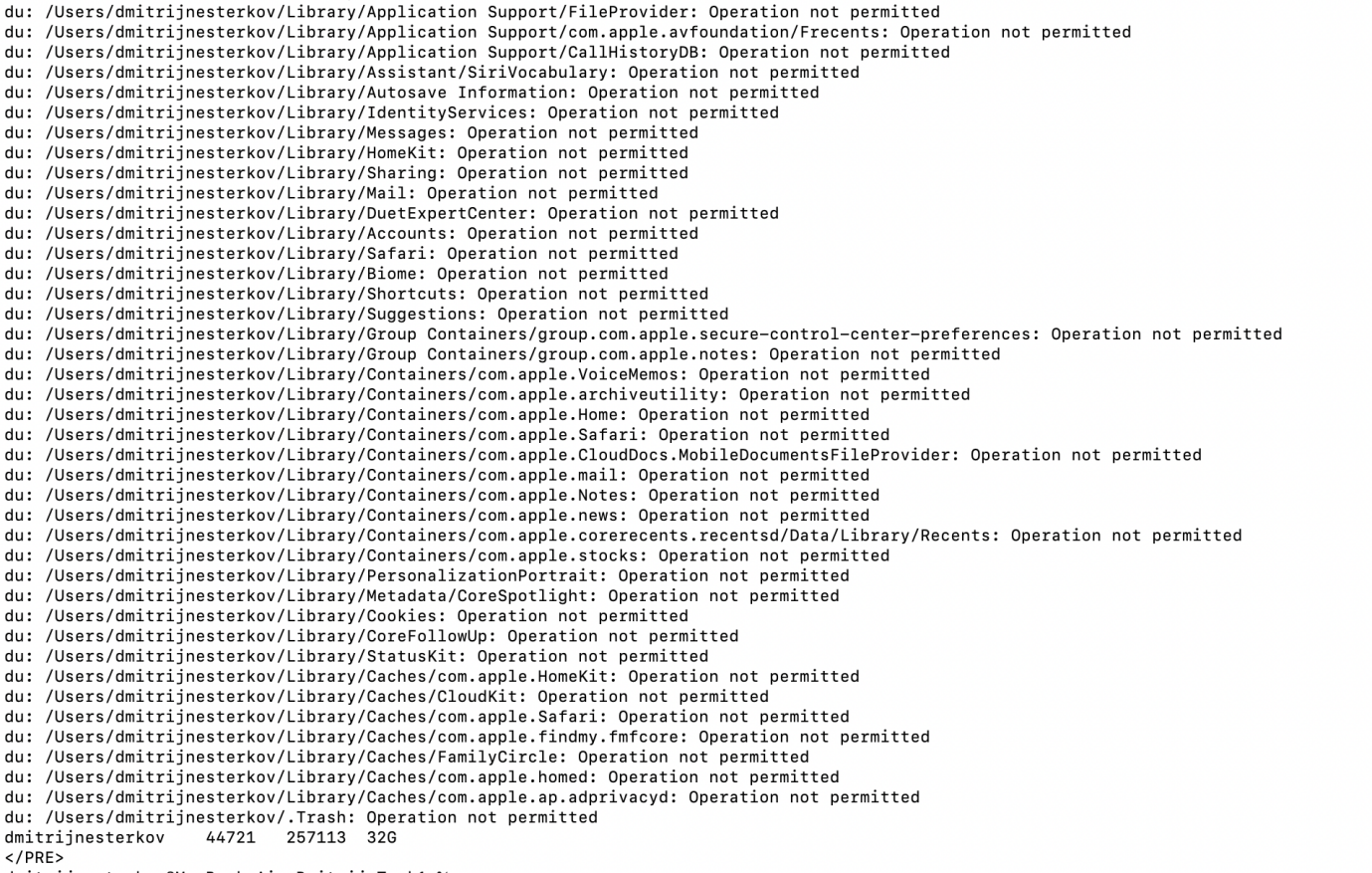


Рисунок 7-8. Копия проекта представелнного в упражнении №3.

**Задание 3**

**#!/bin/bash**

**report\_home\_space () {**

**local format="%-15s%-15s%-15s\n"**

**local i dir\_list total\_files total\_dirs total\_size user\_name**

**if [[ $(id -u) -eq 0 ]]; then**

**dir\_list="/Users/\*"**

**user\_name="All Users"**

**else**

**dir\_list="$HOME"**

**user\_name="$USER"**

**fi**

**output="Home Space Utilization ($user\_name)\n"**

**for i in $dir\_list; do**

**total\_files=$(find "$i" -type f | wc -l)**

**total\_dirs=$(find "$i" -type d | wc -l)**

**total\_size=$(du -sh "$i" | cut -f 1)**

**output+="$i\n"**

**output+=$(printf "$format" "Dirs" "Files" "Size")**

**output+=$(printf "$format" "----" "-----" "----")**

**output+=$(printf "$format" "$total\_dirs" "$total\_files" "$total\_size")**

**done**

**# Используем Zenity для отображения вывода**

**zenity --text-info --width=400 --title="Отчет о дисковом пространстве" --filename=/dev/stdin <<< "$output"**

**}**

**report\_home\_space**

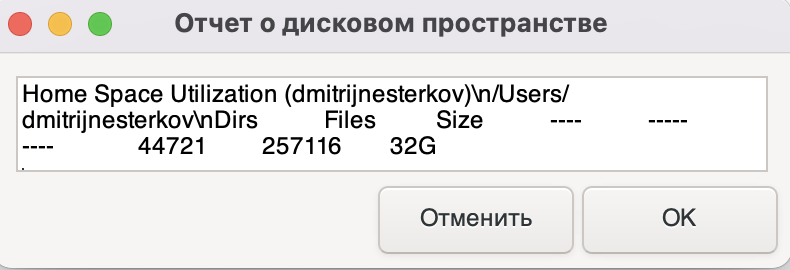
****

Рисунок 8. Сценарий выполненного в задании №2 создан графическое диалоговое окно.

**Контрольные вопросы**

Что такое функции командной оболочки?

Функции командной оболочки (shell functions) - это блоки кода внутри командной оболочки (например, Bash, Zsh, или других), которые могут быть определены один раз и затем вызваны многократно внутри скриптов или интерактивных сессий.

Укажите какими интернет-источниками вы пользовались при изучении и ризработке графического диалога.

chatGPT и claude.ai