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* Филанович Д.В.

Писарик А.С.

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.

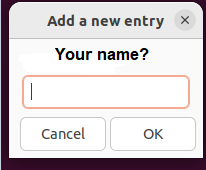
**Решение**

1.Пишем скрипт с двумя вложенными диалогами типа YesNo сразу вместе с элементами для пункта 2.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

2.Запрашиваем на ввод имя человека



3.Запрашиваем у пользователя информацию о его професии.



4.Реализовываем систему ответа на выбранный вариант



**2.**

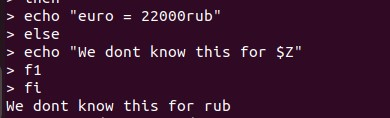
1. Прописываем скрипт и спрашиваем у пользователя валюту, с которой он хочет работать.



Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

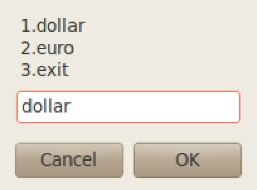
1. При вводе euro получаем котировку, при вводе незаданной валюты, получаем реализованное сообщение о том, что введена не так валюты



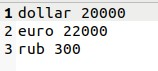
**3.**

1. Делаем оконное меню для валют и сразу реализовываем опцию выхода Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание



1. Создаем отдельное окно для выхода и добавочное окно катировки





**Задание 2.**

6. Создайте копию проекта из упражнения 3.

7. Добавьте туда файловый вывод

**4.** Создаем копию проекта

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**5.** Добавляем туда файловый вывод



**Задание 3.**

8. Довести проект до совершенства, добавив выбор функций

9. Сделать так, чтобы информация хранилась в html файле и открывалась в браузере

**8.**

1. Добавляем домашнюю страницу для выбора функций

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

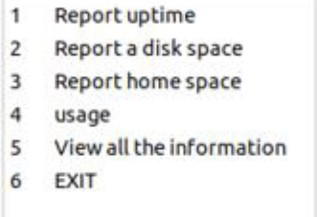
1. Добавляем выбор базовых функций



1. Реализовываем интерфейс для вывод в html документ



1. Получаем итоговое меню для выбора функций



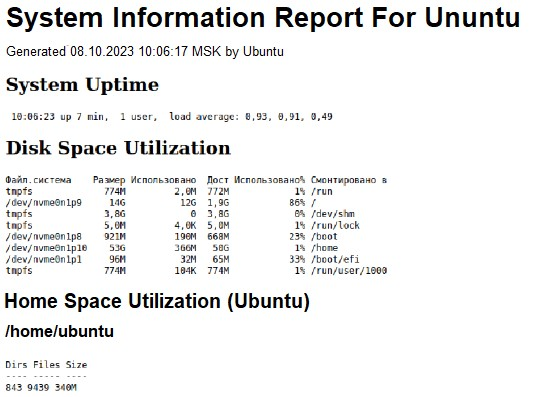
1. Подтверждаем запись в html файл



1. Открываем файл в браузере по нашему пути



1. Наблюдаем за итоговым результатом в браузере



Контрольные вопросы:

1. **Что такое переменные и константы, чем они отличаются? Как их различает командная оболочка?**

Переменные и константы - это символические имена, которые используются для хранения значений в программировании.

Основное отличие между переменными и константами заключается в том, что значение переменной может изменяться в течение выполнения программы, тогда как значение константы остается постоянным и неизменным.

Командная оболочка различает переменные и константы по способу их определения и использования. В большинстве командных оболочек переменные обычно определяются с помощью оператора присваивания, например:

variable=5

Константы, с другой стороны, часто определяются при помощи ключевых слов или специальных функций, которые указывают, что значение не должно изменяться. В некоторых командных оболочках можно создать константу, используя оператор readonly, например:

readonly constant=10

Командная оболочка также может предоставлять специальные синтаксические конструкции для работы с переменными и константами, такие как подстановка значений переменных, операции присваивания и другие функции для работы с данными.

1. **Что такое область видимости переменных? Какие типы области видимости переменных существуют?**

Область видимости переменных - это часть программы, в которой переменная может быть использована или доступна. Область видимости определяет контекст, в котором переменная существует и может быть обращена.

Существует несколько типов области видимости переменных:

1. Глобальная область видимости: Переменная, объявленная в глобальной области видимости, доступна из любой части программы, включая все функции и блоки кода. Глобальные переменные могут быть использованы и изменены в любом месте программы. Они обычно объявляются в начале программы и сохраняют свои значения на протяжении всего времени выполнения программы.

2. Локальная область видимости: Переменная, объявленная внутри функции или блока кода, имеет локальную область видимости. Локальные переменные видимы только в пределах функции или блока, в котором они объявлены. Они могут быть использованы и изменены только внутри той же функции или блока. При выполнении блоков кода для циклов, условий или других структур, локальная область видимости ограничивается соответствующим блоком.

3. Параметры функций: Параметры функций также имеют область видимости, которая ограничена телом функции. Они являются локальными переменными, которые принимают значения аргументов, переданных функции при ее вызове.

Область видимости переменных влияет на доступность и значение переменных в разных частях программы, а также на то, как они взаимодействуют друг с другом. Правильное управление областью видимости помогает избежать конфликтов и позволяет программистам создавать более структурированный и надежный код.

1. **Какое типичное соглашение о написании имен переменных и констант принято программистами? Для чего это делается?**

Типичное соглашение о написании имен переменных и констант, которое принято среди программистов, называется стилем именования переменных и следует определенным правилам. Некоторые распространенные соглашения включают:

1. Имена переменных обычно записываются в нижнем регистре (например, count, name) и могут содержать буквы, цифры и символ подчеркивания \_. Пробелы не разрешены.

2. Для разделения слов в именах переменных обычно используется принятый способ называемый "snake\_case", где слова разделяются символом подчеркивания (например, first\_name, total\_count).

3. Имена констант традиционно записываются в верхнем регистре (например, MAX\_COUNT, PI) и также могут содержать буквы, цифры и символ подчеркивания.

4. Если переменная является локальной и используется только внутри функции или блока кода, то может использоваться некоторое соглашение об использовании "lowerCamelCase", где первое слово начинается с маленькой буквы, а каждое следующее слово начинается с заглавной буквы (например, totalCount, firstName).

Соглашения о именах переменных и констант используются для улучшения читаемости кода и облегчения его понимания. Они помогают сделать код более ясным, согласованным и удобным для сопровождения. Стиль именования переменных отлично подходит, чтобы отличить переменные от констант, понять, что эти символы имеют смысл и соблюдать единообразие во всем коде программы.