

Северо-Кавказский федеральный университет
Институт математики и информационных технологий

ОТЧЕТ
о выполнении лабораторной работы №5
по дисциплине
«Основы Программной Инженерии»

Выполнил:

Ботвинкин Никита Сергеевич

студент 2 курса, ПИЖ-б-о-21-1 группы
бакалавриата «Программная инженерия»
очной формы обучения

Ставрополь, 2023

СКРИНШОТЫ

```
PS C:\git\2_4lab> python Individualnoe.py add data.json --name="ботвинкин" --group="пиж" --marks=52525
PS C:\git\2_4lab> python Individualnoe.py add data.json --name="альбомов" --group="ивт" --marks=33333
PS C:\git\2_4lab> python Individualnoe.py show_marks data.json
+-----+-----+
| ботвинкин | пиж |
+-----+-----+
PS C:\git\2_4lab> python Individualnoe.py show data.json
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | ботвинкин | пиж | 5 2 5 2 5 |
| 2 | альбомов | ивт | 3 3 3 3 3 |
+-----+-----+-----+-----+
```

Рисунок 4.1 – Пример успешной работы программы

```
PS C:\git\2_4lab> python Individualnoe.py add data.json --name="альб" --group="ивт" --marks=333282123
ошибка в количестве или значении оценок
PS C:\git\2_4lab> python Individualnoe.py add data.json --name="альб" --group="ивт" --marks=333
ошибка в количестве или значении оценок
PS C:\git\2_4lab> python Individualnoe.py add data.json --name="альб" --group="ивт" --marks=33327
ошибка в количестве или значении оценок
```

Рисунок 4.2 – Пример ошибок в программе

The screenshot shows a code editor displaying a JSON file. The JSON structure is as follows:

```
[  
  {  
    "name": "альб",  
    "group": "ивт",  
    "marks": [  
      3,  
      3,  
      3,  
      2,  
      3  
    ]  
  },  
  {  
    "name": "альб",  
    "group": "ивт",  
    "marks": [  
      3,  
      3,  
      3,  
      2,  
      5  
    ]  
  }]
```

Рисунок 4.3 – Как выглядит файл data.json

Ответы на контрольные вопросы

1. Каково назначение переменных окружения?

С помощью переменных окружения можно создавать универсальные пути для приложений, которые будут работать на любых ПК, независимо от имен пользователей и других параметров

2. Какая информация может храниться в переменных окружения?

Переменные окружения могут хранить в себе текстовую информацию, которую могут использовать запускаемые программы.

3. Как получить доступ к переменным окружения в ОС Windows?

Получить информацию о существующих переменных можно в свойствах системы > дополнительные параметры системы > дополнительно > переменные среды

4. Каково назначение переменных PATH и PATHEXT?

PATH позволяет запускать исполняемые файлы и скрипты, лежащие в определенных каталогах, без указания их точного местоположения

PATHEXT дает возможность не указывать даже расширения файла, если оно прописано в ее значениях

5. Как создать или изменить переменную окружения в Windows?

Создать или изменить переменную можно в дополнительных настройках системы windows в специальном окне указывая ее путь

6. Что представляют собой переменные окружения в ОС Linux?

Переменные окружения в Linux представляют собой набор именованных значений, используемых другими приложениями

Переменные окружения применяются для настройки поведения приложений и работы самой системы.

7. В чем отличие переменных окружения от переменных оболочки?

Переменные оболочки – это переменные которые применяются только к текущему экземпляру оболочки, в то время как переменные окружения доступны в масштабах всей системы.

8. Как вывести значение переменной окружения в Linux?

Наиболее часто используемая команда – printenv. При вызове без аргументов будет выведены все переменные окружения

9. Какие переменные окружения Linux Вам известны?

HOME – путь к домашнему каталогу

USER – текущий пользователь

PWD – текущая директория

OLDPWD – предыдущая рабочая директория

SHELL – путь к оболочке текущего пользователя

EDITOR – заданный по умолчанию редактор

LOGNAME – имя пользователя, используемое для входа в систему

PATH – пути к каталогам для команд

LANG – текущие настройки языка и кодировки

TERM – тип текущего экземпляра термила

MAIL – место хранения почты

LS_COLORS – цвета для выводения объектов

10. Какие переменные оболочки Linux Вам известны?

BASHOPTS – список задействованных параметров оболочки

BASH_VERSION – версия запущенной оболочки bash

COLUMNS – количество столбцов, которые используется для отображения выходных данных

DIRSTACK – стек директорий, к которому можно применять команды pushd popd

HISTFILESIZE – максимальное количество строк для файла истории команд

HOSTNAME – имя текущего хоста

IFS – внутренний разделитель поля в командной строке

PS1 – определяет внешний вид строки приглашения ввода новых команд

PS2 – вторичная строка приглашения

SHELLOPTS – параметры оболочки, которые можно устанавливать с помощью команды set

UID – идентификатор текущего пользователя

11. Как установить переменные оболочки в Linux?

NEW_VAR=""

12. Как установить переменные окружения в Linux?

С помощью команды export

13. Для чего необходимо делать переменные окружения Linux постоянными?

Если вы хотите, чтобы переменная сохранялась после закрытия сеанса оболочки, то необходимо прописать ее в специальном файле: bashrc

14. Для чего используется переменная окружения PYTHONHOME ?

Эта переменная изменяет расположения стандартных библиотек python.

15. Для чего используется переменная окружения PYTHONPATH ?

Эта переменная изменяет путь поиска по умолчанию для файлов модуля.

16. Какие еще переменные окружения используются для управления работой интерпретатора Python?

PYTHONSTARTUP

PYTHONOPTIMIZE

PYTHONBREAKPOINT

PYTHONDEBUG

PYTHONINSPECT

PYTHONUNBUFFERED

PYTHONVERBOSE

PYTHONCASEOK

PYTHONDONTWRITEBYTECODE

PYTHONPYCACHEPREFIX

PYTHONHASHSEED

PYTHONIOENCODING

PYTHONNOUSERSITE

PYTHONUSERBASE

PYTHONWARNINGS
PYTHONFAULTHANDLER
PYTHONTRACEMALLOC
PYTHONPROFILEIMPORTTIME
PYTHONASYNCIODEBUG
PYTHONMALLOC
PYTHONMALLOCSTATS
PYTHONLEGACYWINDOWSFSENCODING
PYTHONLEGACYWINDOWSSTDIO
PYTHONCOERCECLOCALE
PYTHONDEVMODE
PYTHONUTF8
PYTHONWARNDEFAULTENCODING
PYTHONTREADDEBUG
PYTHONDUMPREFS

17. Как осуществляется чтение переменных окружения в программах на языке программирования Python?

Для доступа к переменным окружения используется объект `os.environ`.

18. Как проверить, установлено или нет значение переменной окружения в программах на языке программирования Python?

Можно проверить через оператор `if`

19. Как присвоить значение переменной окружения в программах на языке программирования Python?

Можно установить значение через `os.environ.setdefault`