

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
дисциплины «Анализ данных»
Вариант №22

Выполнил:
Михеева Елена Александровна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с переменными окружения в Python3

Цель: приобретение навыков по работе с переменными окружения с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Отработали программу примера лабораторной работы.

```
(venv) (base) elenamiheeva@MacBook-Pro-Elena programm % python3 example.py -h
usage: workers [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a new worker
    display            Display all workers
    select             Select the workers

options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
(venv) (base) elenamiheeva@MacBook-Pro-Elena programm % python3 example.py add -h
usage: workers add [-h] [-d DATA] -n NAME [-p POST] -y YEAR

options:
  -h, --help            show this help message and exit
  -d DATA, --data DATA The data file name
  -n NAME, --name NAME   The worker's name
  -p POST, --post POST   The worker's post
  -y YEAR, --year YEAR   The year of hiring
(venv) (base) elenamiheeva@MacBook-Pro-Elena programm %
```

Рисунок 1. Результаты работы программы примера

2. Приступили к выполнению индивидуального задания №1: #
Добавьте возможность получения имени файла данных, используя
соответствующую переменную окружения.

```

import argparse
import json
import sys
import os.path
import sys

def add_students(students, name, group_number, performance):
    """
    Функция для добавления нового студента в список.
    Записывает в файл students.json данные о студенте,
    включая группу и успеваемость,
    создает новую запись, и добавляет ее в список списка студентов,
    сохраняя по файлу.
    """
    students.append(
        {
            "name": name,
            "group_number": group_number,
            "performance": performance
        }
    )
    return students

def list_students(students):
    """
    Функция для вывода списка студентов.
    Выводит рабочее представление списка,
    включая имена, фамилии,
    номер группы и успеваемость.
    """
    if students:
        line = "{:<-10} {:<-10} {:<-10} {:<-10} {}".format(
            "Имя",
            "Группа",
            "Успеваемость",
            "Фамилия и имя отчество",
            ""
        )
        print(line)
        for student in enumerate(students, 1):
            print(
                "{:<-10} {:<-10} {:<-10} {:<-10} {}".format(
                    str(student),
                    student["name"],
                    student["group_number"],
                    student["performance"],
                    ""
                )
            )
            print(line)
    else:
        print("Список студентов пуст")

def find_students():
    """
    Функция для поиска студентов с номером 2.
    """
    found = []
    for student in students:
        if 2 in student["performance"]:
            found.append(student)
    if not found:

```

Рисунок 2. Код індивідуального завдання №1

Выполнено задание №2: самостоятельно изучите работу с пакетом python-dotenv. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла .env.

```
from datetime import datetime
# coding: utf-8

# (Самостоятельно изучите работу с пакетом python-dotenv)
# Модифицируйте программу задания 1 таким образом, чтобы значения
# необходимых переменных окружения считывались из файла .env

import argparse
import json
import os
import sys
from dotenv import load_dotenv

def add_students(students, name, group_number, performance):
    """
    Функция для добавления нового ученика в список.
    Запрашивает у пользователя фамилию и инициалы студента,
    номер группы и успеваемость,
    сохраняет новую запись и добавляет ее в общий список студентов,
    сортируя по фамилии.
    """
    students.append(
        {
            'name': name,
            'group_number': group_number,
            'performance': performance
        }
    )
    return students

def list_students(students):
    """
    Функция для вывода списка студентов.
    Выводит табличное представление списка,
    включая номер, фамилию,
    номер группы и успеваемость.
    """
    if students:
        line = '{:<4}|{>30}|{>20}|{>20}|{>20}|'.format(
            '№',
            'Фамилия и инициалы',
            'номер группы',
            'успеваемость'
        )
        print(line)
        print(
            [
                '{:<4}|{>30}|{>20}|{>20}|{>20}|'.format(
                    '№',
                    'Фамилия и инициалы',
                    'номер группы',
                    'успеваемость'
                )
            ]
        )
        for idx, student in enumerate(students, 1):
            print(
                [
                    '{:<4}|{>30}|{>20}|{>20}|{>20}|'.format(
                        idx,
                        student.get('name', ''),
                        student.get('group_number', ''),
                        student.get('performance', '')
                    )
                ]
            )
        print(line)
    else:
        print("Список студентов пуст.")

def find_students():
    """
    Функция для поиска студентов с отметкой 2.
    """
    found = []
    for student in students:
        if 2 in student['performance']:
            found.append(student)
    if not found:
        print("Студентов с отметкой 2 не найдено")
    else:
        list_students(found)
    return found

def save_students(file_name, students):
    """
    Сохраняет всех студентов в файл JSON.
    """
    with open(file_name, "w", encoding="utf-8") as fout:
        json.dump(students, fout, ensure_ascii=False, indent=4)

def load_students(file_name):
    """
    Загружает всех студентов из файла JSON.
    """
    if not os.path.exists(file_name):
        return []
    with open(file_name, "r", encoding="utf-8") as fin:
        loaded = json.load(fin)
    return loaded

def main(command_line=None):
    """
    Главная функция программы.
    """
    load_dotenv() # Загрузить переменные окружения из файла .env
    file_parser = argparse.ArgumentParser(add_help=False)
    file_parser.add_argument(
        "-d",
        "--data",
        action="store",
        required=False,
        help="The data file name"
    )
    parser = argparse.ArgumentParser("students")
    parser.add_argument(
        "--help",
        action="version",
        version=f"{prog} 0.1.0"
    )
    subparsers = parser.add_subparsers(dest="command")
    parser.add_argument(
        "add",
        parents=(file_parser,
        help="Add a new student"
        )
    )
    add.add_argument(
        "-n",
        "--name",
        action="store",
        required=True,
        help="Student's name"
    )
    add.add_argument(
        "-g",
        "--group-number",
        action="store",
        required=True,
        help="Student's group number"
    )
    add = subparsers.add_parser(
        "display",
        parents=(file_parser,
        help="Display all students"
        )
    )
    args = parser.parse_args(command_line)
    data_file = args.data
    if not data_file:
        data_file = os.environ.get("STUDENTS_DATA")
    if not data_file:
        print("The data file name is absent, file=sys.stderr)
        sys.exit(1)
    is_dirty = False
    if os.path.exists(data_file):
        students = load_students(data_file)
    else:
        students = []
    if args.command == "add":
        students = add_student(
            students,
            args.name,
            args.group,
            args.performance
        )
        is_dirty = True
    elif args.command == "display":
        list_students(students)
    elif args.command == "find":
        find_students()
    if is_dirty:
        save_students(data_file, students)
    if __name__ == '__main__':
        main()
```

Рисунок 3. Программа задания №2

Вывод: в ходе выполнения работы были приобретены навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.