

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт информатики, математики и робототехники
Кафедра математического и компьютерного моделирования

Лабораторная работа №2: Работа с форматом TOML в Python

ОБУЧАЮЩЕГОСЯ
4 курса группы ПИ-4ИВТ221Б

Санникова Михаила Александровича

Уровень высшего образования:	высшее образование – бакалавриат
Направление подготовки (специальность)	<u>09.03.03 “Прикладная информатика”</u>
Направленность (профиль) программы	<u>Информационные и вычислительные технологии</u>
Дата выполнения	06.10.2025

Постановка задачи:

Создать файл config.toml

Написать функцию для чтения файла

Добавить в файл config.toml:

- Параметр timeout = 30 в таблицу settings
- Новую таблицу paths с параметром home = /usr/local/app
- Массив allowed_ips = ["192.168.1.1", "10.0.0.1"] в таблицу settings

Написать функцию, которая проверяет:

- Существование обязательных параметров (host, user в таблице database)
- Корректность типов данных (host — целое число, debug — булево значение)
- Наличие и корректность ip адресов в allowed_ips

Практическая часть:

Листинг кода с комментариями

```
import os
import toml
import re
from functools import wraps

import toml

def main():
    filename = 'config.toml'

    create_initial_toml(filename)
    modify_toml_file(filename)

    data = read_toml_file(filename)
    if not is_data_valid(data):
        print('Данные некорректны!')
        return

    pretty_print(data)

def create_initial_toml(filename):
    config = {
        'database': {
            'host': 'localhost',
            'port': 5432,
            'user': 'admin',
        },
        'settings': {
            'debug': True,
            'log_level': 'INFO',
        },
    }
    with open(filename, "w", encoding="utf-8") as f:
        toml.dump(config, f)

def check_file_exists(func):
    @wraps(func)
    def wrapper(file_path):
        if not os.path.exists(file_path):
            print(f'Не существует файла с заданным путем: {file_path}')
            return
        return func(file_path)

    return wrapper

@check_file_exists
def read_toml_file(path):
    with open(path, 'rb') as f:
        return toml.load(f)
```

```

def modify_toml_file(filename='config.toml'):
    config = read_toml_file(filename)

    config['settings']['timeout'] = 30
    config['settings']['allowed_ips'] = ["192.168.1.1", "10.0.0.1"]
    config['paths'] = {
        'home': '/usr/local/app'
    }

    with open(filename, "w", encoding="utf-8") as f:
        toml.dump(config, f)

def is_data_valid(data):
    return (
        data["database"].get("host")
        and data["database"].get("user")
        and isinstance(data["database"]["port"], int)
        and isinstance(data["settings"]["debug"], bool)
        and is_ips_valid(data["settings"]["allowed_ips"])
    )

def is_ips_valid(allowed_ips):
    if not allowed_ips:
        print("Ошибка: Массив allowed_ips пуст")
        return False

    ipv4_pattern = r'^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$'

    for i, ip in enumerate(allowed_ips):
        if not isinstance(ip, str):
            print(f"Ошибка: IP-адрес должен быть строкой, получен {type(ip)}: {ip}")
            return False

        ip = ip.strip()
        if not ip:
            print(f"Ошибка: Пустой IP-адрес в позиции {i}")
            return False

        if not re.match(ipv4_pattern, ip):
            print(f"Ошибка: Неверный формат IP-адреса: {ip}")
            return False

    return True

def pretty_print(data):
    for table_name, table_content in data.items():
        print(f"\nТаблица: {table_name}")
        for key, value in table_content.items():
            print(f"\t{key} = {value} (тип: {type(value).__name__})")

if __name__ == '__main__':

```

```
main()
```

Скриншоты выполнения программы

Таблица: database

host = localhost (тип: str)

port = 5432 (тип: int)

user = admin (тип: str)

Таблица: settings

debug = True (тип: bool)

log_level = INFO (тип: str)

timeout = 30 (тип: int)

allowed_ips = ['192.168.1.1', '10.0.0.1'] (тип: list)

Таблица: paths

home = /usr/local/app (тип: str)