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:  Направление подготовки (специальность)  Направленность (профиль)  программы  Дата выполнения | высшее образование – бакалавриат  09.03.03 “Прикладная информатика”  Информационные и вычислительные технологии  06.10.2025 |
|  |  |
|  |  |

УФА – 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* tomllib  
*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* tomllib.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

